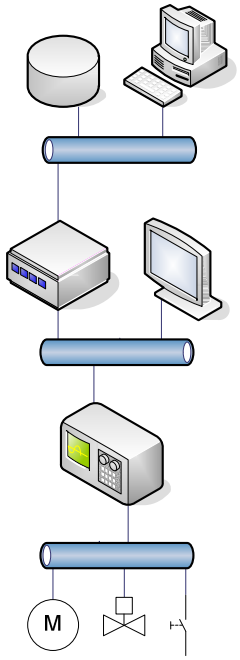
Sorin Larionescu

TEORIA SISTEMELOR



Management

Automatizare

Procese

MATRIX ROM
Bucureşti 2006

Referent ştiinţific: **prof. univ. dr. ing. Sorin Caluianu**

Lucrarea a fost avizată pentru publicare în Şedinţa Catedrei de Electrotehnică a UTCB din 15.11.2005

Tehnoredactarea: autorul

CUPRINS

1. INTRODUCERE....9

2. TEORIA GENERALĂ A SISTEMELOR 10

2.1. SISTEME 11
2.1.1. Elemente....11
2.1.2. Relații....11
2.1.3. Scopuri....11
2.2. DINAMICA SISTEMELOR 12
2.2.1. Variabilele de stare....12
2.2.2. Perturbații....12
2.2.3. Întreținere, apărare, creștere....12
2.3. MEDIUL EXTERIOR SISTEMULUI 12
2.4. STRUCTURA SISTEMULUI 12
2.5. MODELELE SISTEMELOR 12
2.5.1. Definirea modelării sistemice 13
2.5.2. Proprietățile modelării....13
2.5.3. Valoarea modelelor....13
2.6. EXEMPLE DE SISTEME....14
2.6.1. Sistemele liniare....14
2.6.2. Sisteme fizice....14
2.6.3. Sisteme cu variabile de stare....14
2.6.4. Sisteme cu evenimente discrete 14
2.6.5. Sisteme cu eşantionare....14
2.6.6. Sisteme de conducere automată....15
2.6.7. Sisteme informatice....15
2.6.8. Sisteme adaptive....15
2.6.9. Sisteme inteligente....16
2.6.10. Sisteme de domotică....16
2.7. APLICAȚII ALE TEORIEI SISTEMELOR 16
2.7.1. Proiectarea....16
2.7.2. Documentarea....16
2.7.3. Comunicarea....16
2.7.4. Luarea deciziilor....17
2.8. SOFTWARE SPECIALIZAT PENTRU TEORIA SISTEMELOR....17

3. SISTEME FIZICE CU PARAMETRII CONCENTRAȚI....18

3.1. MODELUL MATEMATIC AL SISTEMULUI FIZIC. 20
3.2. MODELE STRUCTURALE DE TIP REȚEA ALE SISTEMULUI FIZIC....22
3.3. MODELELE FUNCȚIONALE ALE SISTEMULUI FIZIC. 23
3.3.1. Metoda generală Kelvin. 23
3.3.2. Programul pentru simularea sistemului fizic. 26
3.3.3. Metoda variabilei auxiliare....30

3.4. ENERGIA ŞI PUTEREA 31
3.5. SURSE REALE DE PUTERE 33
3.6. ECUAŢIILE LUI LAGRANGE 34
3.7. MODELE CU VARIABILE DE STARE ALE SISTEMULUI FIZIC 37
3.8. MODELE ÎN DOMENIUL FRECVENŢĂ ALE SISTEMULUI FIZIC 46
3.8.1. Scurt istoric 47
3.8.2. Semnale sinusoidale 47
3.8.3. Semnale periodice. Seria Fourier 55
3.8.4. Semnale impuls. Transformarea Fourier. Transformarea Laplace 56
3.8.5. Semnificaţii fizice în domeniul frecvenţă 62
3.8.6. Răspunsul sistemelor la semnale impuls în domeniul frecvenţă 62

4. SISTEME LOGICE COMBINAŢIONALE 65

4.1. CODURI 66
4.2. ANALIZA SISTEMELOR LOGICE COMBINAŢIONALE 67
4.2.1. Sistem logic combinaţional tip ŞI 68
4.2.2. Sistem logic combinaţional tip SAU 69
4.2.3. Sistem logic combinaţional tip NU 70
4.2.4. Relaţii logice caracteristice sistemelor logice combinaţionale 71
4.2.5. Elemente reale ale sistemelor logice combinaţionale 72
4.3. SINTEZA SISTEMELOR LOGICE COMBINAŢIONALE 73
4.3.1. Metoda formei disjunctive canonice 73
4.3.2. Metoda diagramei Karnaugh 75
4.4. TESTAREA SISTEMELOR LOGICE COMBINAŢIONALE 80
4.5. SISTEME LOGICE COMBINAŢIONALE CU CIRCUITE INTEGRATE 82

5. SISTEME CU EVENIMENTE DISCRETE 84

5.1. MODELAREA SISTEMELOR CU EVENIMENTE DISCRETE 85
5.2. DEFINIREA SISTEMELOR DISCRETE LOGICE. 89
5.3. TIPURI DE REŢELE PETRI. 90
5.3.1. Reţele Petri autonome. 90
5.3.2. Reţele Petri interpretate. 90
5.3.3. Reţele Petri temporizate. 91
5.4. ANALIZA STRUCTURALĂ A SISTEMELOR CU EVENIMENTE DISCRETE. 91
5.4.1. Structuri tip folosite la modelarea cu reţele Petri 91
5.4.2. Reţele Petri ordinare. 92
5.4.3. Reţele Petri pure. 92
5.4.4. Poziţia sursă sau receptor. 92
5.4.5. Tranziţia validată. 92
5.4.6. Tranziţia declanşată. 93
5.4.7. Conflictul structural şi conflictul efectiv al tranziţiilor. 93
5.4.8. Interblocarea prin interpretare. 94
5.5. ANALIZA COMPORTAMENTALĂ A SISTEMELOR CU EVENIMENTE DISCRETE. 94
5.5.1. Ecuaţia de stare. 94
5.5.2. Graful marcajelor accesibile. 95
5.6. PERFORMANŢELE SISTEMELOR CU EVENIMENTE DISCRETE. 95
5.6.1. Reversibilitatea 96
5.6.2. Mărginirea şi siguranţa. 97
5.6.3. Viabilitatea. 99
5.7. SISTEME CU EVENIMENTE DISCRETE ŞI EVOLUŢIE PARALELĂ. 100
5.8. MAŞINA DE STARE. 103
5.9. GRAFCETUL. 105

5.9.1. Trecerea de la reţeaua Petri la grafcet. *105*
5.9.2. Etape, tranziţii şi legături orientate. *106*
5.9.3. Interpretarea tranziţiilor. *107*
5.9.4. Interpretarea etapelor. *107*
5.9.5. Reguli de evoluţie în grafcet. *108*
5.9.6. Structuri folosite la modelarea cu grafcet *108*
5.9.7. Compararea grafcetului cu reţeaua Petri. *109*
5.10. SINTEZA SISTEMELOR DISCRETE LOGICE. 110
5.11. IMPLEMENTAREA SISTEMELOR DISCRETE LOGICE. 111
5.12. AUTOMATE ELEMENTARE CU CONTACTE ŞI RELEU. 112
5.12.1. Analiza structurală. *113*
5.12.2. Analiza comportamentală. *114*
5.12.3. Automatul elementar cu basculare. *115*
5.12.4. Automatul elementar cu prioritate la oprire. *119*
5.12.5. Automatul elementar cu prioritate la pornire. *122*
5.12.6. Automatul elementar cu neschimbarea stării. *125*
5.12.7. Automatul elementar pentru reglarea bipoziţională. *126*
5.13. AUTOMATE ELEMENTARE CU CIRCUITE INTEGRATE 130
5.13.1. Bistabilul RS. *131*
5.13.2. Bistabilele SR şi SRC. Sincronizarea. *133*
5.13.3. Bistabilul SCR Master – Slave. *134*
5.13.4. Bistabilul JKC. *134*
5.13.5. Bistabilul D. *135*
5.13.6. Bistabilul T. *135*
5.14. SINTEZA AUTOMATELOR IMPLEMENTATE CU CONTACTE ŞI RELEE 136
5.14.1. Schema tehnologică şi schema bloc *136*
5.14.2. Caietul de sarcini al automatului *140*
5.14.3. Analiza structurală şi comportamentală *140*
5.14.4. Sinteza automatului cu contacte şi relee. *141*
5.14.5. Implementarea automatului cu contacte şi relee. *144*
5.15. PROIECTAREA AUTOMATELOR IMPLEMENTATE CU BISTABILE 146
5.15.1. Sinteza şi implementarea automatului cu bistabile tip D *146*
5.15.2. Sinteza şi implementarea automatului cu bistabile JKC. *149*
5.15.3. Problema iniţializării sistemului discret logic. *150*
5.16. PROIECTAREA AUTOMATELOR IMPLEMENTATE CU APL 154
5.16.1. Programarea automatelor programabile logice *154*
5.16.2. Metoda Grafcet *157*
5.16.3. Metoda listei de instrucţiuni *157*
5.16.4. Metoda schemei desfăşurate *161*
5.17. TRANSFORMAREA APL ÎN AUTOMAT CU CONTACTE ŞI RELEE 162

6. SISTEME DE REGLARE AUTOMATĂ 164

6.1. SISTEME CU CONDUCERE MANUALĂ ŞI AUTOMATĂ 165
6.2. REGULATOARE PID CONTINUE 174
6.2.1. Compensatorul PID analogic. *176*
6.2.2. Regulatorul PID cu două grade de libertate *176*
6.2.3. Regulatorul PI cu impulsuri modulate în durată *178*
6.3. PROIECTAREA SISTEMELOR AUTOMATE LINIARE 180
6.3.1. Analiza sistemelor automate *182*
6.3.2. Schema bloc a sistemului automat *184*
6.3.3. Sensibilitatea *185*
6.3.4. Sensibilitatea complementară *185*

6.3.5. Locul rădăcinilor *186*
6.3.6. Ieşirea, eroarea si comanda instalaţiei automate *187*
6.3.7. Măsuri ale erorii *187*
6.3.8. Metoda clasică de proiectare *188*
6.3.9. Metoda modernă de proiectare *190*
6.3.10. Criteriu de stabilitate robustă *192*
6.3.11. Criteriu de performanţe robuste *196*
6.3.12. Compensarea perturbaţiilor *200*
6.3.13. Exemplu de proiectare *204*
6.3.14. Exemplu de simulare *209*
6.4. PROBLEMA TIMPULUI MORT ÎN PROIECTAREA ASISTATĂ DE CALCULATOR 214
6.5. SISTEME AUTOMATE CU REGULATOARE CU MODEL INTERN 220
6.5.1. Definirea regulatorului cu model intern *221*
6.5.2. Proiectarea regulatorului cu model intern *222*
6.5.3. Exemplu de proiectare *224*
6.6. SISTEME ADAPTIVE 229
6.6.1. Sisteme adaptive cu autoacordare *229*
6.6.2. Sisteme adaptive cu model de referinţă *230*

7. SISTEME CU EŞANTIONARE 238

7.1. DISCRETIZAREA ÎN AMPLITUDINE 239
7.2. EŞANTIONAREA 239
7.3. MULTIPLEXAREA 241
7.4. TRANSFORMAREA Z A SEMNALELOR EŞANTIONATE 241
7.5. FUNCŢIA DE TRANSFER ÎN DOMENIUL Z 243
7.6. FUNCŢIA DE TRANSFER ÎN Z ECHIVALENTĂ 245
7.7. COMPENSATORUL PID NUMERIC 247
7.8. INSTRUMENTUL VIRTUAL PID. 247
7.9. MODELUL ARMAX 248
7.10. SISTEME CU SEMNALE ALEATOARE 252
7.10.1. Media şi varianţa *253*
7.10.2. Covarianţa şi corelaţia *254*
7.10.3. Eşantionarea statistică şi estimarea parametrilor *254*
7.10.4. Un exemplu de semnal aleator *255*
7.10.5. Intervale de încredere *256*
7.10.6. Filtrarea *257*
7.10.7. Analiza semnalelor aleatoare *258*
7.10.8. Funcţia de autocorelaţie. *258*
7.10.9. Densitatea spectrală de putere. *259*
7.10.10. Răspunsul sistemelor la semnale aleatorii. *261*
7.11. EŞANTIONAREA ÎN DOMENIUL FRECVENŢĂ. 262
7.12. REGULILE DE EŞANTIONARE ALE KITSAS 265
7.13. IDENTIFICAREA SISTEMELOR 265

8. SISTEME INTELIGENTE CU LOGICA FUZZY 268

8.1.1. Mulţimi fuzzy. *269*
8.1.2. Logica fuzzy. *274*
8.1.3. Sisteme de conducere automată bazate pe logica fuzzy. *277*
8.1.4. Avantajele conducerii fuzzy si domenii de aplicare. *286*
8.1.5. Sinteza algoritmului de conducere fuzzy pentru o instalaţie. *287*
8.1.6. Simularea sistemelor bazate pe logica fuzzy cu ajutorul KitSAS. *292*
8.2. REGULATORUL PID FUZZY. 295

9. SISTEME DE DOMOTICĂ ŞI IMOTICĂ.....299

9.1. SISTEME INFORMATICE 303
9.1.1. Sisteme informatice la nivelul de planificare.....303
9.1.2. Sisteme informatice la nivelul de management.....304
9.1.3. Sisteme informatice la nivelul de cunoaştere.....304
9.1.4. Sisteme informatice la nivelul operaţional.....304
9.2. SISTEME DE DOMOTICĂ CU STRUCTURI PARTICULARE.....305
9.2.1. Sistem de domotică pentru conducerea automată.....305
9.2.2. Sistem de domotică pentru gestiunea clădirii - BMS.....305
9.2.3. Conducerea tip SCADA/HMI.....306
9.2.4. Sistem de domotică pentru securitatea clădirii.....306
9.2.5. Sistem de domotică pentru siguranţa clădirii.....306
9.2.6. Sistem de domotică pentru gestiunea energiei.....306
9.3. PROTOCOALELE SISTEMULUI DE DOMOTICĂ.....307
9.3.1. Protocoalele CEN TC247 şi BACnet.....307
9.3.2. Protocoalele RS-232, RS-422, RS-485 şi HART.....310
9.3.3. Protocoale la nivelul linie - legătura de date.....312
9.3.4. Nivelele reţea, transport, sesiune, prezentare şi aplicaţie.....313
9.3.5. Protocolul PROFIBUS.....314
9.3.6. Protocolul EIB Instabus.....315
9.4. SISTEME DE CONDUCERE ÎN CAZ DE PERICOL.....316
9.4.1. Centrale de conducere în caz de pericol.....318
9.4.2. Traductoare pentru sisteme de securitate la efracţie.....319
9.4.3. Detectoare pasive în infraroşu - PIR.....319
9.4.4. Monitoare video de mişcare.....319
9.4.5. Controlere de acces.....319
9.4.6. Sisteme de identificare cu frecvenţă radio.....320
9.4.7. Traductoare pentru sisteme de siguranţă la incendiu.....320
9.4.8. Butoane manuale de semnalizare.....322
9.4.9. Detectoare de temperatură.....323
9.4.10. Detectoare de fum cu ionizare.....325
9.4.11. Detectoare de fum optice.....326
9.4.12. Detectoare de fum fotoelectrice cu obturarea luminii.....326
9.4.13. Detectoare de fum fotoelectrice cu difuzarea luminii.....326
9.4.14. Sistemul MicroSAM.....326
9.5. SISTEME DE ECHIPAMENTE.....329

10. MEDII DE DEZVOLTARE A APLICAŢIILOR.....333

10.1. KITSAS.....333
10.2. SCILAB / SCICOS.....340
10.3. SCILAB / CONTROLKIT.....343
10.3.1. Instalarea.....343
10.3.2. Ajutor.....343
10.3.3. Pornirea şi utilizarea Control Kitului.....344
10.3.4. Proiectarea minimală cu ajutorul Control Kitului.....344
10.3.5. Proiectarea cu asigurarea stabilităţii şi robusteţii.....345
10.4. DYNAST.....347
10.4.1. Modelul matematic neliniar explicit.....348
10.4.2. Instrucţiuni pentru elaborarea programelor DYNAST.....349
10.4.3. Modelul matematic neliniar implicit.....350
10.4.4. Modelul structural sub forma de reţea.....352

10.4.5. Modelul structural neliniar *354*
10.4.6. Modelul funcțional. *355*
10.5. SIMULATORUL APL TRILOGI 356
10.5.1. Simulatorul Trilogi Tl4 pentru DOS *356*
10.5.2. Încărcarea unui program existent. *357*
10.5.3. Elaborarea unei scheme desfăşurate noi. *357*
10.5.4. Simularea funcționarii unui automat programabil. *357*
10.5.5. Configurarea automatului programabil. *358*
10.5.6. Introducerea unui comentariu. *358*
10.5.7. Editarea unei scheme desfăşurate. *358*
10.5.8. Prezentarea tabelului de configurare si a schemei desfăşurate. *359*
10.5.9. Un exemplu de program *359*
10.5.10. Simulatorul Tl5 pentru Windows *363*
10.6. SIMULATORUL APL EASY RELAY 363

11. BIBLIOGRAFIE 364

1. Introducere

Teoria sistemelor este o disciplină ştiinţifică care încearcă o abordare unitară a metodelor de cunoaştere şi acţiune asupra lumii *obiective* şi *subiective*. După cum se vede ambiţiile teoriei sistemelor sunt foarte mari. În continuare se va face o prezentare generală a teoriei sistemelor şi se va exemplifica aplicarea ei pentru sistemele care apar la instalaţiile, echipamentele şi serviciile din clădiri, grupuri de clădiri şi locuinţe. Se vor considera sistemele fizice cu parametrii concentraţi, sistemele logice combinaţionale, sistemele cu evenimente discrete, sistemele de reglare automată, sistemele cu eşantionare, sistemele informatice, sistemele adaptive, sistemele inteligente cu logică fuzzy şi sistemele de domotică şi imotică. Modelele sistemelor prezentate sunt de tip matematic, structural sau funcţional. Se insistă asupra *structurii* sistemelor şi asupra *metodelor de analiză* a lor. Terminologia şi notaţiile adoptate sunt cele folosite în câteva dintre programele de referinţă în domeniu: *Matlab, Scilab*[1] şi *Dynast*[2]. Programele gratuite *Dynast* şi *Scilab/Scicos* sunt prezentate pe scurt şi folosite la aplicaţiile[3] din lucrare. Se prezintă şi se folosesc şi programele *KitSAS*[4] şi *Scilab/ControlKit*[5] elaborate special de autor pentru simularea, analiza şi proiectarea sistemelor. Pentru sistemele discrete logice se foloseşte simulatorul gratuit de automate programabile *Trilogi*[6].

Elementele sistemelor pot fi atât obiecte fizice cât şi date, simboluri, concepte, etc. În această lucrare se studiază în cea mai mare parte modelele sistemelor formate din obiecte fizice. Aceasta înseamnă că o categorie largă de sisteme, care pot fi în legătură cu activităţile din clădiri, grupuri de clădiri şi locuinţe, nu sunt prezentate. Printre acestea se numără şi sistemele informatice bazate pe calculatoare sau reţele de calculatoare şi a căror elemente sunt datele.

În clădiri şi locuinţe sistemele sunt definite la trei niveluri: management, automatizare şi câmp (procese). În lucrare sistemele vor fi prezentate în ordine ierarhică inversă.

---

[1] Un mediu de dezvoltare a aplicaţiilor sistemice bazat pe ***modele funcţionale*** compatibil în mare măsură cu Matlab. Poate fi descărcat gratuit de la adresa www.scilab.org. În lucrare se foloseşte versiunea ***Scilab 3.1.1.***

[2] Un mediu de dezvoltare a aplicaţiilor sistemice bazat pe ***modele structurale. Versiunea 3.1.1 Dynast Shell*** off line este folosită în lucrare. Ea este gratuită în varianta cu limitări pentru studenţi. Poate fi descărcată de pe internet de la adresa: *http://virtual.cvut.cz/dyn/download/public*. Versiunea *on line* de la aceiaşi adresă este gratuită şi nu are limitări. Programul poate fi folosit împreună cu Matlab.

[3] Aplicaţiile pot fi descărcate de la www.geocities.com/larionescu/.

[4] Versiunea ***KitSAS 6.12*** se poate descărca gratuit de pe internet de la una dintre adresele www.geocities.com/larionescu/ sau *www.larionescu.home.ro*.

[5] Un program elaborat în Scilab 3.1.1 pentru proiectarea asistată de calculator a sistemelor de reglare automată. ***Versiunea ControlKit 3.0*** este accesibilă gratuit la adresele www.geocities.com/larionescu sau www.larionescu.home.ro.

[6] O versiune ***TL4.1 DOS*** sau o versiune ***TL5 Windows*** se poate descărca gratuit de la adresa *www.tri-plc.com/trilogi.htm*.

2. Teoria generală a sistemelor

"Nu se poate poruncii naturii decât supunându-te ei"
"A cunoaşte adevărat înseamnă a cunoaşte prin cauze[7]"
Francis Bacon, 1561 – 1626

Teoria sistemelor este formată dintr-o mulţime de *concepte[8], metode* şi *principii[9]* şi constituie una dintre căile de cunoaştere şi acţionare asupra lumii *obiective* şi *subiective*. Dacă pe lângă teoria sistemelor se consideră şi mulţimea formată din aplicaţiile ei, diferite ipoteze, preferinţe de aparate şi echipamente, standarde şi norme naţionale şi internaţionale, criterii de evaluare a soluţiilor, lucrări publicate şi programe de calculator, personalităţi recunoscute din domeniu, se obţine *paradigma sistemică[10]*. Există paradigme pentru sisteme sociale, economice, fizice, tehnologice, biologice, psihice, informaţionale, informatice, organizaţionale, etc.

Cuvîntul *sistem* este un concept care desemnează o parte a lumii în care trăim. Teoria sistemelor este privită de diferiţi autori drept o teorie logică – matematică, o teorie formală, un mod de gîndire, un mod de percepere a lumii, o metateorie sau un metalimbaj. In continuare vom înţelege prin teoria sistemelor o modalitate de *cunoaştere* şi *acţiune* asupra lumii *obiective* şi *subiective* şi vom aborda aplicaţii din domeniul[11] instalaţiilor[12], echipamenmtelor[13] şi serviciilor[14] din clădiri, grupuri de clădiri şi locuinţe. În acest sens se pot definii, în legătură cu clădirile şi locuinţele, diferite sisteme fizice (*mecanice, termice, fluidice, electrice*), sisteme de conducere automată, sisteme informaţionale de documentare, sisteme informatice bazate pe reţele de calculatoare, sisteme de fabricaţie, sisteme de organizare, etc. Aceste sisteme sunt studiate în mod tradiţional prin teorii şi paradigme specifice. Paradigma sistemică prezentată în continuare încearcă să pună în evidenţă aspectele comune ale acestor teorii facilitând astfel cunoaşterea, comunicarea, documentarea şi studiul interacţiunii între diferitele tipuri de subsisteme din clădiri şi locuinţe.

---

[7] Aceste aforisme baconiene constitue punctul de plecare al gândirii sistemice care presupune renunţarea la cunoaştere în scopul desăvârşiri ei. Conceptele de cutie neagră (black box) şi dipol se bazează pe acest mod de abordare a cunoaşterii. O alternativă la gândirea sistemică o constitue cunoaşterea nelimitată, până la atingerea esenţei lucrurilor.

[8] Idei abstracte.

[9] Teze, legi fundamentale unanim acceptate. De exemplu principiul terţului exclus care guvernează cunoaşterea şi acţiunea umană.

[10] În sensul lui Thomas Kuhn din cadrul teoriei revoluţiilor ştiinţifice.

[11] Aceste aplicaţii implică atât lumea obiectivă a fenomenelor fizice cât şi lumea subiectivă a percepţiilor subiecţilor umani din clădiri şi locuinţe.

[12] Instalaţii de aer condiţionat, încălzire, frig, electrice şi de iluminat.

[13] Echipamente pentru ascensoare, gătit, spălat, televiziune, telefonie, etc.

[14] Aceste servicii pe care le oferă clădirea se referă, printre altele, la accesul în clădire (de exemplu scara de serviciu), confortul termic şi luminos, securitatea la furt şi jaf, siguranţa la foc, etc.

Lumea obiectivă şi subiectivă poate fi abordată pentru cunoaştere şi acţiune într-o manieră diferită de cea definită de paradigma sistemică. De exemplu, modelele matematice ale fenomenelor fizice obţinute sub formă de ecuaţii integro – diferenţiale sunt de tip sistemic. Modelele matematice ale aceloraşi fenomene fizice obţinute sub forma unor ecuaţii diferenţiale cu derivate parţiale nu sunt de tip sistemic.

2.1. Sisteme

Un sistem este definit drept o mulţime de elemente, o mulţime de relaţii între elemente şi o mulţime de scopuri. Determinarea elementelor, relaţiilor şi scopurilor nu este simplă deoarece depinde, într-o oarecare măsură, de observatorul sistemului. Această observaţie reflectă *principiul incertitudinii* care afirmă că atunci când definim un sistem ne referim la observaţiile noastre care sunt mai degrabă produsul nostru decât al sistemului observat.

Comportarea elementelor sistemului este limitată de relaţiile dintre ele. Să considerăm un grup de copii pe un teren de joacă. Câţiva aleargă, alţii sar, câţiva sunt antrenaţi într-un joc şi mai mulţi stau şi privesc. Există aici un sistem? Da, sunt mai multe. Putem izola sistemele observând restricţiile impuse comportării elementelor. Jocul, de exemplu, impune restricţii copiilor care participă la el. Numai anumite mişcări sunt permise la un moment dat. Jocul, definit ca un sistem, are un scop: câştigarea jocului. Deci sistemul numit joc constă dintr-o îmbinare de elemente, relaţii şi scopuri

2.1.1. Elemente

Elementele sistemului sunt formate din obiecte, idei sau evenimente. Adeseori elementele sunt ele însăşi sisteme. Dacă nu se poate sau nu se doreşte descompunerea mai departe în subsisteme elementul este reprezentat printr-un bloc (cutie neagră) sau dipol. Această metodă se numeşte *analiză funcţională*.

2.1.2. Relaţii

Elementele sistemului sunt în legătură unul cu altul în diferite moduri. Unele relaţii sunt fizice, altele sunt spaţiale sau temporale[15], altele sunt logice[16]. Principiul incertitudinii se aplică şi la relaţii. Un observator economist va insista pe relaţiile financiare, un specialist în marketing evidenţiază relaţiile de cumpărare, iar un informatician se ocupă de schimbul de informaţii. Relaţiile contribuie la atingerea scopurilor sistemului prin limitarea comportării elementelor.

2.1.3. Scopuri

În cazul scopurilor sistemului principiul incertitudinii acţionează în sens invers. Este imposibil să se cunoască scopurile din interiorul sistemului. Aceasta înseamnă că elementele sistemului, dacă sunt *conştiente* şi *inteligente*, pot în cel mai bun caz să tragă concluzii numai pe termen scurt în legătură cu cauzele pentru care comportarea lor este limitată. Numai un observator din exterior poate constata

[15] În legătura dintre ele intervine timpul.
[16] Un element este necesar pentru celălalt.

interacțiunea sistemului cu mediul său exterior şi înțelege de ce elementele se comportă într-un anumit fel.

2.2. Dinamica sistemelor

Sistemele se modifică în timp datorită mediului înconjurător şi relațiilor între elemente.

2.2.1. Variabilele de stare

Variabilele interne care definesc dinamica sistemului se numesc variabilele de stare. Acestea sunt, de obicei, legate de elementele care depozitează energie, materie sau informație.

2.2.2. Perturbații

Variabilele externe care determină schimbarea sistemului se numesc perturbații. aceste pot fi *aditive*, atunci când se adună la variabilele de stare, sau *parametrice,* dacă schimbă parametrii elementelor.

2.2.3. Întreținere, apărare, creştere

La interacțiunea cu mediul exterior sistemul reacționează în trei moduri. *Întreținerea* repară o relație sau un element. *Apărarea* distruge un inamic iar *creşterea* asimilează un subsistem.

Sistemele automate pot repara, într-o oarecare măsură, relațiile şi pot micşora efectele perturbațiilor parametrice.

Perturbațiile pot provoca *instabilitatea* sistemului.

2.3. Mediul exterior sistemului

Sistemul poate evolua într-un mediu *ordonat, dezordonat* sau *turbulent.* Evenimentele din mediul exterior turbulent pot apare sau nu[17]. Dacă mediul este dezordonat evenimentele au diferite probabilități de apariție, totuşi suma lor este unu. Intr-un mediu turbulent nu se cunosc toate evenimentele şi probabilitățile lor de apariție.

2.4. Structura sistemului

Câteva dintre structurile sistemelor sunt: legătura directă, structura cu reacție pozitivă sau negativă, structura cibernetică şi structura cu învățare. Primele două tipuri sunt utilizate frecvent în practica instalațiilor, echipamentelor şi serviciilor oferite de clădiri şi locuințe. Sistemele cu structura directă pot supraviețuii în medii ordonate iar cele care au structura cu reacție fac față şi în medii dezordonate. Sistemele din medii turbulente trebuie să aibă o structură cibernetică. Structura cu învățare este necesară în sistemele foarte complexe.

2.5. Modelele sistemelor

[17] Probabilitatea de apariție este zero sau unu.

Sistemele din natură[18], numite *sisteme de bază*[19], pot fi modelate, adică se pot construii sisteme mai simple , numite *sisteme model*[20], a căror comportare să fie aproximativ aceiaşi. Un sistem poate avea mai multe modele. Studiul comportării sistemelor model se numeşte *simulare*. Schema tehnologică, schema bloc, schema desfăşurată şi reţeaua Petri şi alte desene sunt sisteme model.

2.5.1. Definirea modelării sistemice[21]

- Baza şi modelul sunt sisteme.
- Modelul are cel mult tot atâtea elemente decât baza. Deci este mai simplu.
- Modelul este mai simplu şi în ceea ce priveşte relaţiile dintre elemente. Numărul lor este mai mic sau egal cu cele ale bazei.
- Orice este adevărat din comportarea modelului este adevărat şi referitor la comportarea bazei.

2.5.2. Proprietăţile modelării

- *Nesimetria*. Modelarea merge numai într-o direcţie. Dacă A modelează pe B atunci B nu poate modela pe B.
- *Reflexivitatea*. Orice model este un model al lui însuşi.
- *Tranzitivitatea*. Dacă B este un model al lui C şi A este un model al lui B atunci A modelează şi pe C.
- *Netransferabilitatea*. Două modele ale aceleiaşi baze nu sunt în mod necesar echivalente.
- *Complexitatea redusă*. Simplificarea obţinută prin modelare se obţine prin gruparea elementelor similare sau eliminarea elementelor nesemnificative.
- *Nepartajarea*. Un model a unei parţi a bazei nu este un model a întregii baze.

2.5.3. Valoarea modelelor

Valoarea modelelor pentru comunicare, documentare şi luarea deciziilor este dată de următoarele trei criterii.

- *Consistenţa.* Modelele trebuie să fie întocmite conform standardelor şi normelor în vigoare astfel încât utilizatori să le poată folosi uşor.
- *Generalitatea.* Toate aspectele importante ale sistemului de bază trebuie modelate.
- *Validitatea.* Toate concluziile obţinute pe model trebuie să fie adevărate şi pentru sistemul de bază.

[18] Există controverse cu privire la natura sistemică a realităţii.
[19] Sau mai simplu simplu, *baze.*
[20] Sau mai simplu, *modele.*
[21] Există şi alte tipuri de modelări. De exemplu modelarea matematică sau fizică.

2.6. Exemple de sisteme

Paradigma sistemică se aplică în foarte multe domenii ale ştiinţei şi tehnicii. Câteva tipuri de sisteme sunt prezentate în continuare.

2.6.1. Sistemele liniare

Sistemele în care se aplică principiul superpoziţiei se numesc sisteme liniare. Principiul superpoziţiei afirmă că răspunsul sistemului la intrări aditive este egal cu suma răspunsurilor la fiecare intrare în parte.

2.6.2. Sisteme fizice

Sistemele fizice cu elemente cu parametrii concentraţi au numai trei tipuri de elemente: disipatoare, acumulatoare prin intermediul variabilei longitudinale şi acumulatoare prin intermediul variabilelor transversale. Relaţiile dintre elemente sunt determinate de transferul de putere sau energie dintre elementele sale. Modelele elementelor sunt în acest caz de tip dipol iar modelul sistemului este de tip structural[22]. Proprietăţile structurii sistemelor fizice determinate de geometria sistemului sunt studiate de *topologie*.

2.6.3. Sisteme cu variabile de stare

Aceste sisteme sunt nişte modele funcţionale ale sistemelor fizice sau ale sistemelor automate. În afară de variabilele de intrare şi ieşire se definesc şi variabilele de stare. Alegerea variabilelor de stare este, într-o oarecare măsură, arbitrară.

2.6.4. Sisteme cu evenimente discrete

Evenimentele discrete apar în mod frecvent la organizarea serviciilor oferite de clădiri şi locuinţe, în funcţionarea instalaţiilor, echipamentelor informatice şi a atelierelor de producţie. Din această cauză sistemele cu evenimente discrete sunt foarte utile în luarea deciziilor referitoare la diferite aspecte ale activităţii inginereşti, economice şi organizatorice legate de clădiri şi locuinţe. Sistemele discrete logice sunt cazuri particulare, mai simple, de sisteme cu evenimente discrete şi sunt folosite intensiv la conducerea automată a instalaţiilor şi echipamentelor din clădiri şi locuinţe.

2.6.5. Sisteme cu eşantionare

Pentru multe sisteme se folosesc *modele discrete cu eşantionare*. În această categorie intră sistemele organizaţionale, economice, financiare, sociale sau sistemele tehnice cu calculatoare. Aceste modelele au o arie largă de aplicare deoarece presupun prezenţa la intrarea în sistem a unui semnal determinist combinat cu un semnal aleator. In felul acesta modelele pot fi folosite nu numai la studiul sistemelor dar şi la studiul semnalelor.

[22] Pune în evidenţă structura.

2.6.6. Sisteme de conducere automată

Elementele sistemelor de conducere sunt procesele[23], traductoarele, elementele de execuţie, automatele, regulatoarele şi controlerele[24]. Structura sistemelor de conducere este determinată de transferul de informaţie şi poate fi de tip feedforward, feedback negativ, feedback pozitiv sau combinaţii. Elementele sunt modelate, în general, prin blocuri[25] iar modelul sistemului este de tip funcţional[26]. Sistemele automate pot urmări unul sau mai multe scopuri: a) urmărirea unui program[27], b) înlăturarea efectului perturbaţiilor prin reglare, c) compensarea regimurilor tranzitorii, d) insensibilizarea la variaţia parametrilor sistemului. Bucla cu reacţie negativă, reţeaua Petri sau grafcetul sunt modele ale unor sisteme de conducere automată.

2.6.7. Sisteme informatice

Elementele unui sistem informatic îl constituie datele[28] şi echipamentele care le prelucrează. Relaţiile dintre date îl constituie tranzacţiile. Scopul unui sistem informatic îl constituie prelucrarea datelor. Sistemele expert, bazele de date relaţionale, sistemele de gestiune a clădirilor[29], sistemele de securitate[30], sistemele de siguranţă[31] la foc sau sistemele de prelucrare a tranzacţiilor financiare sunt exemple de sisteme informatice.

Mai multe calculatoare conectate în reţea formează un sistem informatic. Relaţiile dintre elementele sistemului (calculatoarele) pot fi definite la nivel hardware sau software şi formează diferite structuri: stea, magistrală, arbore, etc. Scopul sistemelor informatice îl constituie comunicarea dintre calculatoare. Internetul şi intranetul sunt sisteme informatice.

2.6.8. Sisteme adaptive

Sistemele automate bine acordate au calitatea remarcabilă de a fi insensibile la modificări mic ai parametrilor proceselor conduse. Dacă aceşti parametrii se modifică este necesară şi adaptarea regulatoarelor la noua situaţie.

Sistemele adaptive se deosebesc de sistemele automate clasice prin apariţia unei noi bucle.

---

[23] Într-o instalaţie pot avea loc mai multe procese, de natură fizică diferită.

[24] Microcalculatoare specializate.

[25] Cutii negre.

[26] Pune în evidenţă funcţia de transfer de la intrare la ieţire.

[27] Referinţe.

[28] Datele sunt informaţii prelucrate. Legătura dintre date şi informaţii este studiată de către semiotică. În cadrul sistemelor informaţionale datele sunt organizate în fişiere.

[29] BMS, Building management system. Se gestionează energia electrică şi termică, apa caldă, apa rece, numărul orelor de funcţionare a instalaţiilor, etc.

[30] Sisteme de securitate la furt, jaf şi sisteme de control a accesului.

[31] Se gestionează evenimentele de detectare şi alarmare la incendiu.

2.6.9. Sisteme inteligente

Funcţionarea sistemelor în medii ostile, aşa cum este cazul echipamentelor şi instalaţiilor pentru clădiri şi locuinţe, conduce, printre altele, la existenţa unui nivel ridicat de perturbaţii şi zgomote. Dacă zgomotele depăşesc aproximativ 10 – 20 % din valoarea semnalului util identificarea funcţiei de transfer a procesului nu mai este exactă şi conducerea adaptivă nu mai dă rezultate mulţumitoare. În această situaţie se folosesc pentru conducere sisteme cu inteligenţă artificială cum ar fi sistemele expert şi reţelele neurale.

2.6.10. Sisteme de domotică

Elementele sistemului de domotică îl constituie subsistemele fizice, automate, informaţionale şi informatice. Relaţiile dintre elemente, care formează structura sistemului, este dată de protocoale cum ar fi Bacnet sau EIB. Scopul sistemului de domotică îl constituie comunicarea dintre componentele sale.

2.7. Aplicaţii ale teoriei sistemelor

După cum am mai menţionat paradigma sistemică reprezintă o cale de cunoaştere şi acţiune. Câteva dintre domeniile cele mai importante în care poate fi folosită sunt: proiectarea, documentarea, comunicarea, luarea deciziilor, fabricarea, întreţinerea şi altele.

2.7.1. Proiectarea

Multe metode de proiectare[32] a instalaţiilor şi echipamentelor sunt de tip sistemic. Pe lângă acestea trebuiesc menţionate metodele de proiectare pentru ofertare, pentru instalaţii şi echipamente multidisciplinare[33], pentru conducerea automată şi pentru sistemele informatice.

2.7.2. Documentarea

Foarte multe documente în legătură cu clădirile, locuinţele şi serviciile asigurate de acestea sunt de fapt modele sistemice de diferite tipuri. Un exemplu tipic este schema tehnologică cu sau fără aparatura de automatizare. Aceasta pune în evidenţă elemente, relaţiile dintre ele şi este elaborată în scop de documentare.

2.7.3. Comunicarea

Comunicarea între oameni, eventual de specializări diferite, este facilitată de paradigma sistemică. Schemele bloc reprezintă un exemplu de model sistemic larg răspândit în comunicare datorită simplităţii lor. Comunicarea om – calculator în cazul programelor de uz general[34] este bazată tot pe paradigma sistemică.

---

[32] Unele metode de proiectare, de exemplu cele care folosesc modele sub forma ecuaţiilor cu derivate parţiale, nu sunt bazate pe teoria sistemelor.

[33] De exemplu mecano – electric sau termo – electric.

[34] Pentru programele specializate interfaţa om – calculator este, de obicei, de tip meniu.

2.7.4. Luarea deciziilor

Adeseori suntem nevoiți să luăm decizii în legătură cu conducerea manuală sau automată, cumpărarea sau vinderea, aprecierea calității instalațiilor, echipamentelor şi serviciilor, aprecierea confortului, aprecierea soluțiilor de proiectare, etc. Pentru aceasta un ajutor important îl oferă posibilitatea simulării. Programele generale de simulare sunt întocmite pe baze sistemice.

2.8. Software specializat pentru teoria sistemelor

Aplicațiile teoriei sistemelor folosesc diferite *medii de dezvoltare* software care permit *construirea* diferitor tipuri de modele şi *simularea* lor. Dintre aceste menționăm Dynast, Matlab, Scilab şi KitSAS.

3. Sisteme fizice cu parametrii concentraţi

Modelarea, proiectarea, simularea şi conducerea diferitor echipamente care asigură serviciile oferite de clădiri şi locuinţe poate fi făcută în mod unitar, cel puţin într-o primă aproximaţie, folosind o metodă sistemică. Acest fapt este deosebit de important pentru studenţii Facultăţii de Instalaţii care studiază echipamente termice, hidraulice, electrice, electronice, mecanice şi de automatizare. Abordarea sistemică prezintă mai multe avantaje, dintre care se evidenţiază folosirea unei singure metode pentru diferite tipuri de echipamente şi posibilitatea de studiere a interacţiunii obiectelor de natură fizică diferită.

Un *sistem* este definit ca o mulţime de *elemente* între care există o mulţime *relaţii* în vederea realizării unor *scopuri.* Intr-o clădire pot fi definite sisteme fizice, sisteme informaţionale, sisteme intelectuale, sisteme de producţie, sisteme artistice şi altele în diferite scopuri. Instalaţiile şi echipamentele dintr-o clădire pot fi interpretate drept sisteme de tipuri diferite. Foarte cunoscut este sistemul fizic de *elemente cu parametrii concentraţi* cu componente de tip *dipol* şi de tip *sursă* . Dipolii au o bornă de intrare şi o bornă de ieşire ca în *Fig. 3.1*. Mulţimea elementelor (dipoli şi surse) care formează un sistem este obţinută prin *discretizarea spaţială* a proceselor care au loc în instalaţii şi echipamente. Observarea *obiectivă* a proceselor se poate face numai prin *măsurarea* a două tipuri de mărimi dinamice ce caracterizează dipolii şi sursele: mărimi longitudinale $\lambda(t)$ şi mărimi transversale $\tau(t)$. Legătura dintre elementele sistemului se face prin intermediul acestor mărimi. Variabilele fizice longitudinale şi transversale selectate prin *analogie* pentru diferite tipuri de procese sunt prezentate în Tab. 3.1. Alte mărimi longitudinale sau transversale pot fi alese. De exemplu, pentru procesele hidraulice din rezervoare în care viteza fluidului este neglijabilă se foloseşte în locul presiunii p înălţimea h a lichidului pe baza relaţiei $p = \rho g h$. Pentru instalaţiile şi echipamentele în care au loc mai multe procese fizice diferite care se influenţează în mod semnificativ, variabilele longitudinale şi transversale se aleg astfel încât să fie satisfăcută relaţia (3.1).

$$P(t) = \lambda(t).\tau(t) \qquad (3.1)$$

în care P(t) este puterea instantanee.

Mărimile din Tab. 3.1 satisfac condiţia (3.1) cu excepţia mărimilor longitudinale pentru procesele termice şi fluidice. Dacă se alege pentru procesele termice drept mărime longitudinală debitul de entropie iar pentru procesele fluidice debitul de volum, toate variabilele din tabelul Tab. 3.1 satisfac condiţia (3.1).

În afară de dipoli se pot defini într-un sistem fizic prin discretizare spaţială şi alte elemente de tip surse de mărimi longitudinale şi transversale cu valoarea prescrisă cu simbolurile din *Fig. 3.1*. Sursele permit definirea legăturii sistemului cu exteriorul, definirea condiţiilor iniţiale ale sistemului şi reducerea elementelor cu mai multe borne (multipoli) la o combinaţie de dipoli şi surse cu valoarea comandată de mărimile longitudinale sau transversale ale unor dipoli.

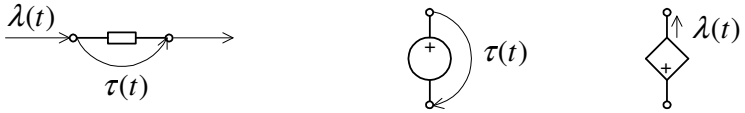


Fig. 3.1 Elementele pasive şi active ale sistemului fizic: dipolul, sursa de variabilă transversală τ şi sursa de variabilă longitudinală λ.

Tab. 3.1 Mărimile longitudinale şi transversale ale elementelor pentru diferite tipuri de procese fizice.

Procesul	**Mărimea longitudinală λ(t)**	**Mărimea transversală τ(t)**
electric	curentul electric i [A]	tensiunea electrică u [V]
termic	debit de căldură q [$J.s^{-1}$]	temperatura θ [K]
fluidic	debit de masă w [$kg.s^{-1}$]	presiunea p [$N.m^{-2}$]
mecanic liniar	forţa f [N]	viteza v [$m.s^{-1}$]
mecanic rotaţional	momentul forţei M [N.m]	viteza unghiulară Ω [$rad.s^{-1}$]

Tab. 3.2 Simboluri de elemente fizice pasive şi ideale

Natura fizica a elementului	***Tipul elementului ideal pasiv***		
	Element disipator de energie	***Element acumulator de energie prin intermediul variabilei longitudinale***	***Element acumulator de energie prin intermediul variabilei transversale***
Mecanica	Amortizor	Resort elestic	Masa
Electrica	Rezistenta	Inductanta	Capacitate
Termica	Rezistenta termica		Capacitate de acumulare a caldurii
Hidraulica	Rezistenta hidraulica	Conducta lunga	Capacitate hidraulica (rezervor)

Pentru fiecare tip de sistem fizic se pot defini *trei* feluri de elemente: elemente disipatoare de energie, elemente acumulatoare de energie prin intermediul variabilei longitudinale şi elemente acumulatoare de energie prin intermediul variabilei transversale Tab. 2.

După stabilirea sistemului şi a elementelor sale componente se pot defini mai multe tipuri de modele ale sistemului. Pentru sistemele fizice cele mai folosite sunt modelul matematic, modelul structural şi modelul funcţional.

3.1. Modelul matematic al sistemului fizic.

Interacţiunea elementelor sistemului fizic este descrisă de relaţii de echilibru şi relaţii de compatibilitate. *Relaţiile de echilibru* se referă la toate variabilele longitudinale care intră sau ies dintr-un nod. Pentru ca sistemul să fie în echilibru suma acestora trebuie să fie egală cu zero. *Relaţiile de compatibilitate* se referă la variabilele transversale ale elementelor care formează o buclă. Suma acestor variabile trebuie să fie egală cu zero. Relaţiile de echilibru şi compatibilitate decurg din structura topologică a sistemului, indiferent de natura fizică a elementelor sale. Pentru fiecare element se poate scrie însă şi o *relaţie fizică*, determinată experimental, între variabila transversală şi variabila longitudinală a elementului.

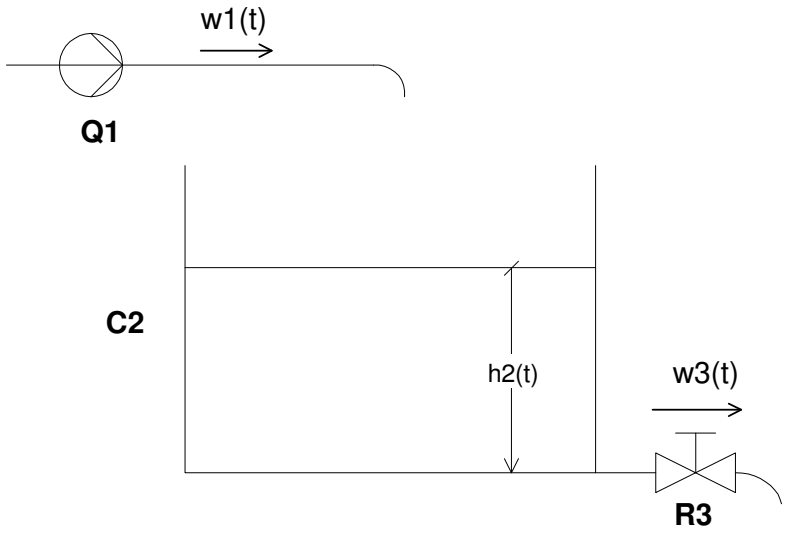


Fig. 3.2. O instalaţie formată dintr-o pompă, un rezervor şi un robinet.

Să prezentăm un exemplu de definire au unui sistem fizic şi a modelului său matematic. Pentru aceasta să considerăm instalaţia din *Fig. 3.2*. In această instalaţie au loc procese electrice, mecanice şi hidraulice. Procesele electrice au loc în motorul pompei. Cuplajul între motor şi pompă, funcţionarea pompei este caracterizată de procese mecanice. Vom ignora aceste procese electrice şi mecanice şi vom defini un sistem fizic hidraulic. Elementele sale componente cu parametrii concentraţi sunt pompa Q1, rezervorul C2 şi robinetul R3. Considerând

variabilele longitudinale şi transversale definite în *Tab. 3.1* pentru componentele sistemelor hidraulice rezultă modelul structural sub formă de reţea hidraulică din *Fig. 3.3*. Se observă că sursa de debit corespunde pompei, capacitatea de acumulare corespunde rezervorului şi rezistenţa hidraulică corespunde robinetului. Reţeaua are două noduri 0 şi 1 şi o buclă formată din elementele C2 şi R3. Relaţia de echilibru a mărimilor longitudinale (debite) în nodul 1 este:

$$w1 = w2 + w3 \tag{3.2}$$

Relaţia de compatibilitate între mărimile transversale (înălţimile hidrodinamice) din buclă este:

$$h2 - h3 = 0 \tag{3.3}$$

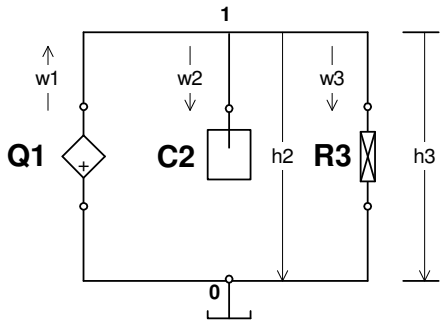


Fig. 3.3. Sistemul fizic cu constante concentrate definit pentru instalaţia pompă – rezervor – robinet.

Relaţia fizică a elementului C2 exprimă faptul că variaţia masei de fluid din rezervor este egală cu debitul de masă w2:

$$w2 = \frac{d(A.\rho.h_2)}{dt} \tag{3.4}$$

în care A este suprafaţa rezervorului iar ρ este densitatea fluidului.

Pentru rezistenţele hidraulice relaţia fizică depinde de regimul de curgere a fluidului. Dacă curgerea prin robinet este turbulentă atunci relaţia fizică pentru rezistenţa R3 este:

$$w3 = \alpha\sqrt{h_3} \tag{3.5}$$

Relaţiile (3.2) .. (3.5) împreună cu condiţiile iniţiale ale variabilelor formează *modelul matematic* al sistemului fizic din *Fig. 3.3*. Cu ajutorul acestui model sistemul poate fi analizat, proiectat şi simulat *fără* să mai fie necesare

structura şi elementele sistemului definite în *Fig. 3.3*. Folosind programe corespunzătoare modelul matematic ne permite să realizăm toate operațiile dorite.

3.2. Modele structurale[35] de tip rețea ale sistemului fizic.

În practica inginerească se definesc, de cele mai multe ori, sisteme fizice *liniare.* Un sistem este liniar dacă i se poate aplica *principiul superpoziției* conform căruia efectul unei sume de cauze aditive este egal cu suma efectelor fiecărei cauze considerate separat. Toate elementele uni sistem liniar trebuie să fie lineare. Sistemele neliniare, cum este de exemplu cel din *Fig. 3.3*, pot fi liniarizate pe porțiuni. Cele trei tipuri de elemente ale unui sistem liniar au aceleaşi trei tipuri de relații fizice între variabila longitudinală şi variabila transversală, indiferent de natura fizică a sistemului.

Elementule pasive disipatoare de energie sunt caracterizate de relația fizică:

$$\tau(t) = \Gamma.\lambda(t) \qquad (3.6)$$

în care Γ este o constantă ce caracterizează rezistența opusă de element la trecerea mărimii longitudinale λ(t).

Elementele pasive acumulatoare de energie prin intermediul mărimii longitudinale λ(t) sunt caracterizate prin relația fizică:

$$\tau(t) = \Phi \frac{d\lambda(t)}{dt} \qquad (3.7)$$

în care Φ este o constantă ce caracterizează acumularea de energie.

Elementele pasive acumulatoare de energie prin intermediul mărimii transversale τ(t) sunt caracterizate prin relația fizică:

$$\lambda(t) = \Psi \frac{d\tau(t)}{dt} \qquad (3.8)$$

în care Ψ este o constantă ce caracterizează acumularea de energie.

Constantele Γ, Φ şi Ψ sunt determinate din considerente fizice, fără să fie necesar modelul matematic. De exemplu, constanta Ψ pentru capacitatea C2 din *Fig. 3.3* este determinată cu relația:

$$C2 = A.\rho \qquad (3.9)$$

în care A este suprafața rezervorului iar ρ este densitatea fluidului.

Modelul structural tip rețea al unui sistem fizic liniar este format dintr-un desen care prezintă mulțimea elementelor componente, valoarea constantelor corespunzătoare şi modul lor de interconectare, adică structura sistemului. Cu un program corespunzător acest desen este introdus în calculator sub formă grafică şi putem analiza, proiecta sau simula sistemul fizic liniar. Nu sunt necesare relațiile matematice care formează modelul matematic al sistemului. Pe baza modelului

[35] Sistemice.

structural şi al definiţiilor (3.5), (3.6) şi (3.8) programul calculează singur modelul matematic care stă la baza operaţiilor efectuate. De exemplu, dacă în *Fig. 3.3* ar fi specificate valorile lui Q1, C2 şi R3 reţeaua respectivă ar forma un model structural. Pentru aceasta relaţia (3.5) trebuie liniarizată şi adusă la forma (3.6).

3.3. Modelele funcţionale[36] ale sistemului fizic.

Adeseori în practica inginerească a sistemelor fizice liniare se doreşte cunoaşterea legăturii dintre un semnal de intrare şi un semnal de ieşire a sistemului. Se determină astfel un model funcţional al sistemului fizic caracterizat printr-o schemă bloc.

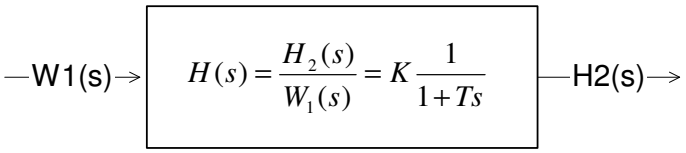


Fig. 3.4. Un model funcţional sub formă de schemă bloc al sistemului din Fig. 3.3

De exemplu, dacă sistemul fizic din *Fig. 3.3* este liniarizat, se poate definii pentru el modelul funcţional sub forma schemei bloc din *Fig. 3.4*. Acest model stabileşte legătura între debitul pompei *w1(t)* şi nivelul fluidului *h2(t)* în rezervor. Se observă în figură că semnalul de intrare, semnalul de ieşire şi funcţia de transfer sunt definite în domeniul frecvenţă cu ajutorul transformatei Laplace. Constantele K şi T ale funcţiei de transfer H(s) sunt determinate nu pe baza modelului matematic sau structural, ci prin alte metode, de exemplu experimental. Constanta de proporţionalitate K precizează modul în care se efectuează transferul semnalului de la intrarea la ieşirea sistemului. Constanta de timp T caracterizează inerţia[37] sistemului. Schema bloc se poate introduce în calculator cu ajutorul unui program corespunzător şi se poate analiza, proiecta sau simula sistemul fizic fără a mai fi necesară cunoaşterea modelului matematic sau structural. Elaborarea modelelor funcţionale

Schema bloc poate fi elaborată din modelul matematic prin mai multe metode. În continuare se prezintă câteva dintre aceste metode şi implementarea modelelor funcţionale obţinute prin programe *KitSAS*[38].

3.3.1. Metoda generală Kelvin.

Metoda Kelvin se aplică în cazul ecuaţiei diferenţiale de forma următoare:

$$\frac{d^2y(t)}{dt^2} = -a_1\frac{dy(t)}{dt} - a_0 y(t) + b_0 u(t) \qquad (3.10)$$

[36] Modele funcţionale de tip sistemic. Există şi alte tipuri de modele funcţionale, de exemplu modelele cu variabile de stare.

[37] mecanică, electrică, termică sau fluidică.

[38] O scurtă introducere în paragraful 10.1

Aceasta este o ecuație diferențială de ordinul doi care reprezintă modelul unui oscilator mecanic sau unei bucle standard de automatizare. Se observă că în membrul doi al ecuației nu există derivate. In primul membru pot exista derivate de orice ordin, s-a ales ordinul doi pentru simplificarea expunerii.

Primul pas al metodei Kelvin de programare constă în separarea derivatei de ordinul cel mai mare:

$$\frac{d^2y(t)}{dt^2} + a_1\frac{dy(t)}{dt} + a_0 y(t) = b_0 u(t) \qquad (3.11)$$

Al doilea pas constă din obținerea variabilei y(t) cu un număr convenabil de integratori **inte**, doi pentru modelul considerat drept exemplu. Cu această ocazie se obține şi prima derivată a lui z(t), *Fig. 3.5*.

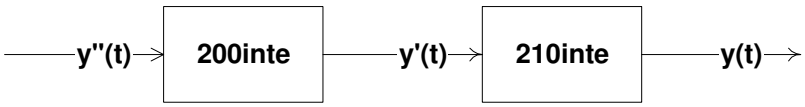


Fig. 3.5. Al doilea pas al metodei Kelvin.

In al treilea pas se observă că partea dreaptă a ecuației se compune din variabila independentă u(t) a cărei variație în timp este cunoscută şi o combinație liniară a lui y(t) cu prima sa derivată. Însă y(t) şi y'(t) au fost obținuți în cel de al doilea pas (*Fig. 3.5*). Se cunosc deci toate datele pentru obținerea expresiei din partea dreaptă a ecuației diferențiale cu ajutorul unui sumator **220suma** ca în *Fig. 3.6*.

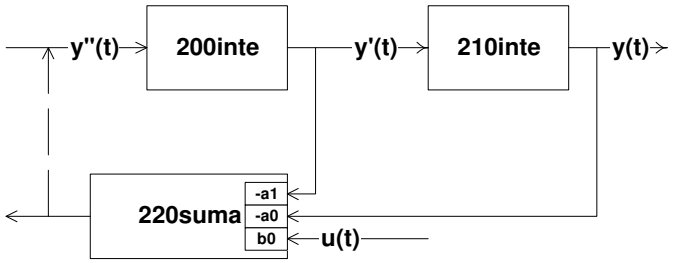


Fig. 3.6. Al treilea pas al metodei Kelvin.

Deoarece expresia din partea dreaptă a ecuației diferențiale este egală cu **y"(t)** cele două semnale pot fi conectate aşa cum se arată de linia punctată din figura *Fig. 3.6*. S-a obținut schema bloc corespunzătoare modelului matematic sub forma ecuației diferențiale de ordinul doi. Programul kit rezulta imediat din *Fig. 3.6*.

Totuşi, blocul de elemente din *Fig. 3.6* nu funcționează, nu poate fi simulat, deoarece nu există generatorul de semnal **u(t)**. Elementele care mai trebuiesc adăugate pentru o simulare comodă vor fi prezentate în paragraful următor

3.3.2. Programul pentru simularea sistemului fizic.

Schema bloc din *Fig. 3.6* este echivalentă programului modelului matematic (3.10). Adeseori însă se doreşte simularea unui sistem fizic pentru care coeficienţii ecuaţiei diferenţiale au anumite semnificaţii.

De exemplu, pentru simularea unui oscilator sau bucle de reglare automată modelul matematic (3.10) are forma:

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n \frac{dy}{dt} + \omega_n^2 y = b_0 u(t) \qquad (3.12)$$

în care ζ este fracţiunea de amortizare critică,

ω_n: pulsaţia naturală a sistemului.

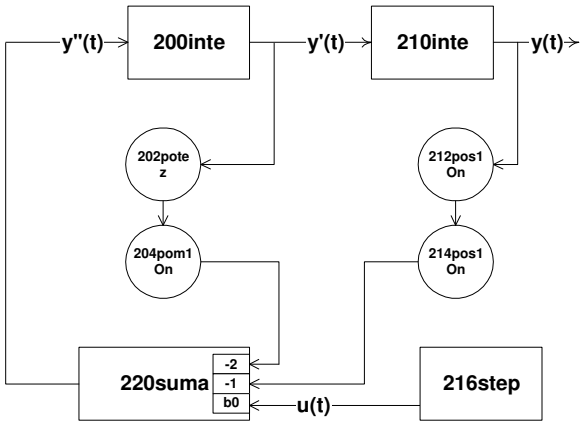


Fig. 3.7. Simularea unui oscilator.

Schema bloc pentru simularea modelului matematic (4.3) a unui oscilator este prezentată în *Fig. 3.7*. Comparativ cu *Fig. 3.6* se observă apariţia a patru potenţiometrii pentru stabilirea parametrilor ζ şi ω_n. Dacă fracţiunea de amortizare critică este specificată cu ajutorul elementului **202pote**, pentru pulsaţia naturală sunt necesari trei potenţiometrii. Este incomod ca o singură valoare numerică să fie introdusă de trei ori. Din această cauză cei trei potenţiometrii 204, 212 şi 214 au fost uniţi în grupa cu numărul unu . Sunt posibile patru astfel de grupe. In această grupă **204pom1** este potenţiometrul master iar **212pos1** şi **214pos1** sunt potenţiometrii slave. Dacă se stabileşte o valoare pentru master automat aceiaşi valoare o iau şi elementele slave. Programul corespunzător schemei bloc din *Fig. 3.7* se găseşte în fişierul Magazie\Demo\Oscilat.s cu următoarele valori numerice pentru parametrii: ζ=0.2 , ω_n=1 şi b_0=1. Conţinutul fişierului este prezentat în continuare. Se observă modul de prezentare a schemei bloc pe coperta programului. Pentru condensarea textului programul este prezentat pe trei coloane.

In program apare şi un element **noop** util pentru modificări ulterioare. Descrierea unui element al blocului este formată din 10 linii şi are următoarea formă:

N = 0
TIP=noop
n1 = 0
n2 = 0
n3 = 0
K1 = .0
K2 = .0
K3 = .0
E = .0
Txt= **********

Primele patru caractere ale fiecărei linii constituie eticheta liniei care conţine numele atributului elementului a cărui valoare este definită în continuare.

Lista 3.1. Coperta şi programul de simulare pentru oscilatorul mecanic.

```
       1           :           Programul           Oscilat.s
y"(t)+(2zOn)y'(t)+(On^2)y(t)=b0.u(t)
       2 :
       3 :    y"-->200inte-->y'-->210inte-->y
       4 :    y'-->202pote-->204pom1-->|-2|220suma-->y"
       5 :    y -->212pos1-->214pos1-->|-1|
       6 :                   216step-->|b0|
       7 :
       8 :    z-->202pote = 0.2
       9 :    On-->204pom1 = 1
      10 :    b0 = 1
      11 :
      12 :    Durata simularii = 20 s
      13 :    Pasul de calcul  = 0.2 s
      14 :    Nr pasi afisare = 5
      15 :    Nr canale de masura = 1
      16 :    Nr elem la care se masoara iesirea y(t) = 210
      17 :
      18 :
      19 :
      20 :
      N  = 200
      TIP=inte
      n1 = 220
      n2 = 0
      n3 = 0
      K1 = .0
      K2 = .0
      K3 = .0
      E  = .0
      Txt=        **********
      N  = 202
      TIP=pote
      n1 = 200
      n2 = 0
      n3 = 0
      K1 = 0.2
      K2 = .0
      K3 = .0
      E  = .0
```

```
Txt=     Fractiunea de amortizare critica zita
N  = 204
TIP=pom1
n1 = 202
n2 = 0
n3 = 0
K1 = 1
K2 = .0
K3 = .0
E  = .0
Txt=     Pulsatia naturala OmegaN
N  = 210
TIP=inte
n1 = 200
n2 = 0
n3 = 0
K1 = .0
K2 = .0
K3 = .0
E  = .0
Txt=     Integrator
N  = 212
TIP=pos1
n1 = 210
n2 = 0
n3 = 0
K1 = 1
K2 = .0
K3 = .0
E  = .0
Txt=     Slave al 204 pentru OmegaN
N  = 214
TIP=pos1
n1 = 212
n2 = 0
n3 = 0
K1 = 1
K2 = .0
K3 = .0
E  = .0
Txt=     Slave al 204 pentru OmegaN
N  = 216
TIP=step
n1 = 0
n2 = 0
n3 = 0
K1 = .0
K2 = .0
K3 = .0
E  = .0
Txt=     Generator treapta unitara
N  = 220
TIP=suma
n1 = 204
n2 = 214
n3 = 216
K1 = -2
K2 = -1
```

```
      K3 = 1
      E  = .0
      Txt=     Sumator
      N  = 230
      TIP=spar
      n1 = 5
      n2 = 5
      n3 = 2
      K1 = 20
      K2 = 0.2
      K3 = .0
      E  = .0
      Txt= K1=T final, K2=Pas calc, n1=nr pas afis, n2= nr
pas esant, n3=metoda
      N  = 240
      TIP=opar
      n1 = 1
      n2 = 210
      n3 = 0
      K1 = .0
      K2 = .0
      K3 = .0
      E  = .0
      Txt=     n1=nr masurari, n2=nr primul canal, n3=nr al
doilea canal
      N  = 0
      TIP=noop
      n1 = 0
      n2 = 0
      n3 = 0
      K1 = .0
      K2 = .0
      K3 = .0
      E  = .0
      Txt=     **********
```

Programul are structura descrisă în paragraful 3.3.2. Faţă de *Fig. 3.7* programul cuprinde trei elemente suplimentare: **230spar, 240opar, 0noop**. Elementele spar şi opar nu prelucrează semnale şi sunt importante prin efectele lor laterale la începutul simulării.

Spar stabileşte parametrii simulării la valorile indicate de constantele sale n1, n2, n3, k1, k2 astfel:

- n1 – Numărul de paşi la care se face afişarea pe videoterminal a rezultatelor.
- n2 – Numărul de paşi la care se face eşantionarea în cazul sistemelor numerice.
- n3 – Tipul sistemului şi metoda numerică utilizată (cod numeric).
- K1 – Durata simulării.
- K2 – Pasul de calcul.

Opar specifică prin constantele sale cum se face măsurarea rezultatelor:

- n1 – Numărul canalelor de măsură. O valoare cuprinsă între 1 şi 5.
- n2 – Numărul elementului la ieşirea căruia este conectat primul canal de măsură. Primele două canale au un statut special putând fi conectate la un osciloscop pentru reprezentarea grafică, la un analizator de frecvenţă sau la un înregistrator numeric pentru redarea exactă a măsurărilor.

- n3 – Numărul elementului la care este conectat al doilea canal de măsură.

Noop nu execută nici o operaţie**.** Prezenţa sa este utilă în cazul unor modificări ulterioare când poate fi înlocuit uşor cu alte elemente.

3.3.3. Metoda variabilei auxiliare

Programarea modelelor matematice cu coeficienţi constanţi şi derivate în membrul drept al ecuaţiei diferenţiale se face prin introducerea unei variabile auxiliare. Pentru exemplificare se consideră modelul matematic:

$$\frac{d^2y(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) = b_1 \frac{du(t)}{dt} + b_0 u(t) \qquad (3.13)$$

Dacă se aplică transformarea Laplace rezultă următoarea funcţie de transfer:

$$H(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} = \frac{Y(s)}{U(s)} \qquad (3.14)$$

Se introduce variabila auxiliară Z(s) prin înmulţirea ei atât cu numărătorul cât şi cu numitorul expresiei (3.14). Se obţine:

$$\frac{Y(s)}{U(s)} = \frac{Z(s).[b_1 s + b_0]}{Z(s).[s^2 + a_1 s + a_0]} \qquad (3.15)$$

Se poate face următoarea identificare:

$$Y(s) = Z(s).[b_1 s + b_0] \qquad (3.16)$$

$$U(s) = Z(s).[s^2 + a_1 s + a_0] \qquad (3.17)$$

Relaţia (3.17) este de forma (3.10) numai că în locul variabilei y(t) apare z(t):

$$\frac{d^2z(t)}{dt^2} + a_1 \frac{dz(t)}{dt} + a_0 z(t) = u(t) \qquad (3.18)$$

Schema bloc corespunzătoare modelului matematic (3.18) este deci de forma din figura *Fig. 3.5*. Pe această schemă se poate măsura z(t) şi prima sa derivată dz(t)/dt adică Z(s) şi sZ(s). Modelul (3.16)în domeniul timp este:

$$y(t) = b_0 z(t) + b_1 \frac{dz(t)}{dt} \qquad (3.19)$$

şi poate fi implementat simplu cu ajutorul unui sumator dacă se cunoaşte y(t) şi derivata sa. Schema bloc finală pentru ecuația (3.13) este prezentată în *Fig. 3.8.*

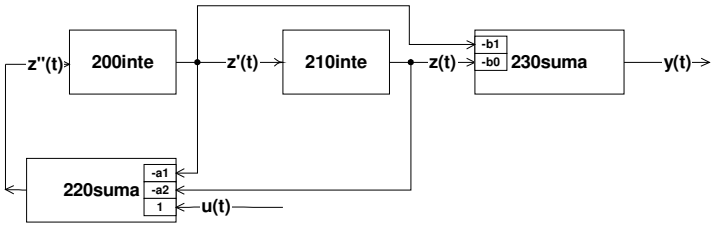


Fig. 3.8. Schema bloc obținută prin metoda de programare a variabilei auxiliare.

3.4. Energia şi puterea

Sistemele tehnice reale cuprind elemente de natura fizica diferita. S-a constatat experimental ca legătura dintre elementele de natura fizica diferita se realizează proporțional cu o mărime măsurabila P(t) numita putere instantanee si egala cu produsul dintre mărimea longitudinala si mărimea transversala din relația (3.1.

Puterea transferata intr-un interval de timp t_1 intre doua elemente constituie chiar energia E produsa, disipata sau consumata:

$$E = \int_0^{t_1} P(t)dt = \int_0^{t_1} \lambda(t) \cdot \tau(t)dt \qquad (3.20)$$

O situație particulara întâlnita frecvent este caracterizata de valorile constante λ_0 si τ_0 ale mărimilor $\lambda(t)$ si $\tau(t)$.In acest caz energia disipata rezulta din (3.20) si (3.6):

$$E = \lambda_0 \cdot \tau_0 \cdot t_1 = \Gamma \cdot \lambda_0^2 \cdot t_1 = \frac{\tau_0^2}{\Gamma} \cdot t_1 \qquad (3.21)$$

Aceste relații sunt simple si ușor de aplicat. Ar fi de dorit ca relații asemănătoare sa fie stabilite si in cazul general in care λ si τ variază in timp. Folosind din nou (3.20) si (3.6) se obține:

$$E = \Gamma \cdot \int_0^{t_1} \lambda^2(t) dt \qquad (3.22)$$

Se impune ca energia sa fie calculata si in acest caz cu ajutorul unei expresii de forma:

$$E = \Gamma \cdot \lambda_{ef}^2 \cdot t_1 \qquad (3.23)$$

in care λ_{ef} se numește valoarea eficace a lui λ(t).

Egalând relațiile (3.22) si (3.23) se stabileşte expresia valorii eficace:

$$\lambda_{ef} = \sqrt{\frac{1}{t_1} \cdot \int_0^{t_1} \lambda^2(t) dt} \qquad (3.24)$$

O expresie asemănătoare se poate calcula si pentru mărimea transversala τ(t).

Pentru cazul particular in care λ(t) si τ(t) se modifica in timp in mod periodic, cu perioada T, atunci in (3.24) se foloseşte:

$$t_1 = T = \frac{2\pi}{\Omega} \qquad (3.25)$$

in care Ω este pulsația variabilei.

In aceasta situație λ(t) si τ(t) pot fi dezvoltate in serie Fourier:

$$\lambda(t) = \lambda_0 + \sum_{k=1}^{\infty} \lambda_k \cdot \cos(k \cdot \Omega \cdot t + \varphi_k) \qquad (3.26)$$

in care λ_k este valoarea maxima a componentelor sinusoidale.

Înlocuind in (3.24) si integrând se deduce:

$$\lambda_{ef}^2 = \lambda_0^2 + \sum_{k=1}^{\infty} \lambda_{kef}^2 \qquad (3.27)$$

adică pătratul valorii eficace este egal cu

suma pătratelor valorilor eficace a tuturor componentelor Fourier. O expresie asemănătoare cu (3.27) se poate calcula si pentru mărimea transversala $\tau(t)$.

Energia acumulata de elemente de tip Φ in intervalul t_1 rezulta din relațiile (3.20) si (3.8):

$$E = \int_0^{t_1} \lambda(t) \cdot \Phi \cdot \frac{d\lambda}{dt} dt = \Phi \cdot \int_{\lambda(0)}^{\lambda(t_1)} \lambda d\lambda = \frac{\Phi \cdot \left[\lambda^2(t_1) - \lambda^2(0)\right]}{2} \qquad (3.28)$$

Energia acumulata de elementele de tip Ψ se calculează asemănător:

$$E = \frac{\Psi \cdot \left[\tau^2(t_1) - \tau^2(0)\right]}{2} \qquad (3.29)$$

3.5. Surse reale de putere

Spre deosebire de sursele din *Fig. 3.1* sursele reale de putere nu pot furniza mărimi longitudinale $\lambda(t)$ sau transversale $\tau(t)$ cu valoarea prescrisa.

Sursele reale de putere pot conține elemente disipative de tip Γ. Ele pot fi reprezentate cu ajutorul surselor ideale de mărime transversala si longitudinala ca in Fig. 3.9:

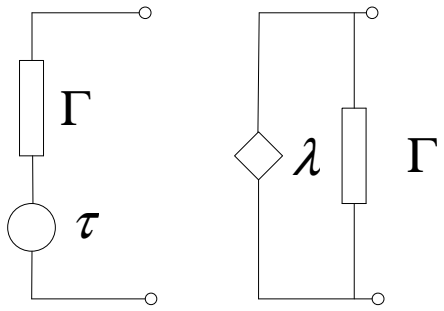


Fig. 3.9 Surse reale de putere cu element disipative

a) Reprezentarea cu ajutorul sursei τ

b) Reprezentarea cu ajutorul sursei $\lambda\Gamma$.

Cele doua scheme din Fig. 3.9 sunt echivalente cu :

$$T = \Lambda \cdot \Gamma \qquad (3.30)$$

Elementele de tip Φ si Ψ cu energie acumulata se comporta la fel cu sursele reale de putere. Schemele lor echivalente sunt prezentate in Fig. 3.10. Mărimile $\lambda(0)$ si $\tau(0)$ reprezintă valorile variabilelor longitudinale si transversale la timpul t=0.Ele se mai numesc condiții inițiale. Energia acumulata de aceste

elemente prin intermediul mărimilor longitudinale $\lambda(t)$ si transversale $\tau(t)$ se calculează cu ajutorul relațiilor (3.28) si (3.29).

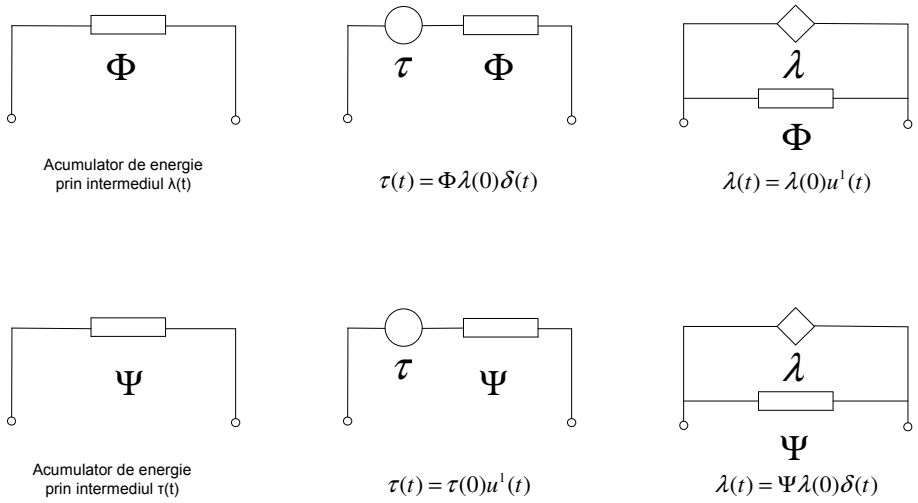


Fig. 3.10 Surse reale de putere cu elemente acumulatoare de energie de tip Φ si de tip Ψ

Sursele echivalente de tip T sau Λ din Fig. 3.10 sunt definite cu ajutorul funcției Dirac $\delta(t)$, al funcției treapta unitate $u^1(t)$ si al condițiilor inițiale $\lambda(0)$ si $\tau(0)$.

3.6. Ecuațiile lui Lagrange

S-au evidențiat pana acum câteva proprietăți importante ale sistemelor fără nici o specificare despre natura fizica a acestora. Rezultatele obținute pot fi aplicate la fel de bine sistemelor mecanice, electrice, termice, hidraulice, etc. In continuare se va stabili in aceeaşi maniera ecuațiile care descriu comportarea in timp (dinamica) a sistemului. Pentru aceasta se presupune ca sistemul are un număr suficient de borne astfel încât mărimile măsurate sa descrie complet starea sistemului. Un sistem care se bucura de aceasta proprietate se numeşte observabil. Sistemul poate fi observat numai prin intermediul mărimilor longitudinale $\lambda(t)$ sau transversale $\tau(t)$. Numărul minim de mărimi prin care sistemul este observabil formează un grup de coordonate generalizate.

Se considera in continuare cazul mai simplu al sistemelor care nu interacționează cu nici un sistem exterior, deci sunt formate numai din elemente acumulatoare de energie de tip Φ si Ψ.Aceste sisteme se numesc conservative deoarece îşi păstrează energia inițială acumulata in elementele de tip Φ si Ψ.Energia totala E a unui sistem conservativ este egala cu suma energiilor înmagazinate in fiecare dintre elementele de tip Φ si Ψ:

$$E = E_\Phi + E_\Psi \tag{3.31}$$

Cu ajutorul energiilor acumulate in elementele care formează sistemul conservativ se poate defini o alta caracteristica importanta a sistemului, numita si funcția lui Lagrange (L):

$$L = E_\Phi - E_\Psi \tag{3.32}$$

în care

E_Φ = energia totala acumulata in elementele de tip Φ;

E_Ψ = energia totala acumulata in elementele de tip Ψ.

Experimental s-a verificat următorul *principiu al lui Hamilton*: între doua momente de timp t_1 si t_2 sistemul îşi modifica starea astfel încât integrala următoare sa aibă o valoare minima:

$$S = \int_{t_1}^{t_2} L dt = \int_{t_1}^{t_2} (E_\Phi - E_\Psi) dt = \min \tag{3.33}$$

Din punctul de vedere sistemic adoptat pana acum relația precedenta poate fi interpretata ca o reflectare a faptului ca in evoluția sa in timp sistemul nu are o preferința fata de elementele de tip Φ sau Ψ.Aceasta este foarte important deoarece definirea elementelor de tip Φ sau de tip Ψ ale sistemului a fost făcuta numai pe baza unor considerente metodologice.

Energia E_Φ a elementelor de tip Φ se poate determina cu relația (3.28) in funcție de mărimea longitudinala λ, iar energia E_Ψ a elementelor de tip Ψ se poate determina cu relația (3.29) in funcție de mărimea transversala τ.Vom considera $\lambda(0)=0$ si $\tau(0)=0$.

Un grup de coordonate generalizate utilizate frecvent este format din variabilele $\mu(t)$ si derivata sa definite astfel:

$\dot{\mu}(t)$

$$\dot{\mu}(t) = \tau(t) \tag{3.34}$$

Dar

$$\tau(t) = \Phi \cdot \frac{d\lambda(t)}{dt} = \Phi \cdot \dot{\lambda}(t) \tag{3.35}$$

Din (3.34) şi (3.35) rezulta:

$$\mu(t) = \Phi \cdot \lambda(t) \tag{3.36}$$

Folosind noul sistem de coordonate generalizate

$$\mu(t), \dot{\mu}(t)$$

energiile acumulate in condiții inițiale nule rezulta din (3.28) si (3.29):

$$E_{\Phi} = \frac{\Phi \cdot \lambda^2(t)}{2} = \frac{\mu^2(t)}{2\Phi} \tag{3.37}$$

$$E_{\Psi} = \frac{\Psi \cdot \dot{\mu}^2(t)}{2} \tag{3.38}$$

Pentru un sistem format din doua elemente, unul de tip Φ si altul de tip Ψ condiția de minim (3.33) poate fi formulata in modul următor:

$$dS = d\int_{t_1}^{t_2} L(\mu, \dot{\mu}, t)dt = \int_{t_1}^{t_2} (\frac{\partial L}{\partial \mu} d\mu + \frac{\partial L}{\partial \dot{\mu}} d\dot{\mu})dt = 0 \tag{3.39}$$

Se observa ca

$$d\dot{\mu} = \frac{d}{dt} d\mu \tag{3.40}$$

Al doilea termen din integrala (3.39) se integrează prin părți. Rezultă:

$$dS = \left[\frac{\partial L}{\partial \dot{\mu}} d\mu \right]_{t_1}^{t_2} + \int_{t_1}^{t_2} \left(\frac{\partial L}{\partial \mu} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\mu}} \right) d\mu \cdot dt = 0 \tag{3.41}$$

Calculul acestei expresii se face păstrând limitele intervalului de mişcare constante. Deci

$$\begin{aligned} &\mu(t_1) = const \\ &\mu(t_2) = const \\ &d\mu(t_1) = const \\ &d\mu(t_2) = const \end{aligned} \tag{3.42}$$

In aceste condiții relația (3.41) se anulează numai daca:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mu}} - \frac{\partial L}{\partial \mu} = 0 \qquad (3.43)$$

Atunci când se determina mai multe coordonate generalizate μ_I ale sistemului se obțin ecuațiile lui Lagrange:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\mu}_i} - \frac{\partial L}{\partial \mu_i} = 0 \qquad i = 1,2,\ldots,n \qquad (3.44)$$

De exemplu, pentru sistemul conservativ cu doua elemente, unul de tip Φ si altul de tip Ψ, energiile sunt date de relațiile (3.37) si (3.38) iar funcția lui Lagrange (3.32) devine:

$$L = \frac{\mu^2}{2\Phi} - \frac{\Psi \cdot \dot{\mu}^2}{2} \qquad (3.45)$$

Introducând pe L in (3.43) se obține:

$$-\Psi \cdot \ddot{\mu} - \frac{\mu}{\Phi} = 0 \qquad (3.46)$$

Particularizând aceasta relație pentru un sistem mecanic avem:

$\dot{\mu} = \tau = \dot{x}$ viteza relativa;

$\mu = x$ deplasarea relativa;

$\Phi = \frac{1}{k}$ k = constanta de elasticitate;

$\Psi = m$ m = masa

$$m \cdot \ddot{x} + k \cdot x = 0 \qquad (3.47)$$

Aceasta este ecuația diferențial binecunoscuta a unui sistem mecanic format dintr-un resort cu constanta de elasticitate k si un corp de masa m.

3.7. Modele cu variabile de stare ale sistemului fizic

Instalațiile (sistemele de echipamente) sunt studiate cel mai adesea cu ajutorul sistemelor fizice cu parametrii concentrați. In acest caz modelul matematic corespunzător este un sistem de ecuații diferențiale. Modelul sistemic cel mai folosit este insa sistemul de ecuații diferențiale de ordinul întâi, numit sistem de ecuații de stare. Obținerea ecuațiilor de stare este ilustrata printr-un exemplu tipic. In urma scrierii relațiilor fizice a componentelor unui sistem cu parametrii

concentraţi şi a aplicării condiţiilor de echilibru şi de compatibilitate, se obţine următoarea ecuaţie diferenţiala:

$$\frac{d^3y}{dt^3}+a_2\cdot\frac{d^2y}{dt^2}+a_1\cdot\frac{dy}{dt}+a_0\cdot y=c_2\cdot\frac{d^2u}{dt^2}+c_1\cdot\frac{du}{dt}+c_0\cdot u \qquad (3.48)$$

in care
u(t) este excitaţia sistemului iar y(t) este răspunsul sistemului.
Se considera variabila intermediara z(t) conform următoarei scheme bloc:

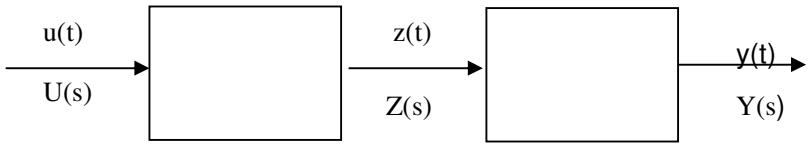


Fig. 3.11 Schema bloc a instalaţiei cu explicitatea intrării u(t), ieşirii y(t) şi a variabilei intermediare z(t)

Din (3.48) şi *Fig. 3.11* rezulta:

$$\frac{Y(s)}{U(s)}=\frac{c_2s^2+c_1s+c_0}{s^3+a_2s^2+a_1s+a_0}=\frac{Y(s)}{Z(s)}\cdot\frac{Z(s)}{U(s)} \qquad (3.49)$$

In (3.49) se pot face următoarele identificări:

$$\frac{Y(s)}{Z(s)}=c_2s^2+c_1s+c_0 \qquad (3.50)$$

$$\frac{Z(s)}{U(s)}=\frac{1}{s^3+a_2s^2+a_1s+a_0} \qquad (3.51)$$

Trecând din nou in domeniul timp, relaţiile (3.50) şi) devin:

$$y=c_2\cdot\frac{d^2z}{dt^2}+c_1\cdot\frac{dz}{dt}+c_0\cdot z \qquad (3.52)$$

$$\frac{d^3z}{dt^3} + a_2 \cdot \frac{d^2z}{dt^2} + a_1 \cdot \frac{dz}{dt} + a_0 \cdot z = u \tag{3.53}$$

In (3.52) şi (3.53) se aleg variabilele de stare x_1, x_2 şi x_3 astfel:

$$x_1 = z \tag{3.54}$$

$$x_2 = \frac{dz}{dt} = \frac{dx_1}{dt} \tag{3.55}$$

$$x_3 = \frac{d^2z}{dt^2} = \frac{dx_2}{dt} \tag{3.56}$$

Ecuația (3.56) poate fi pusa sub forma:

$$\frac{d^3z}{dt^3} = \frac{dx_3}{dt} \tag{3.57}$$

Cu ajutorul relațiilor [(3.53)…(3.57)] se poate elimina variabila z şi se obține următorul sistem de ecuații de stare:

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 & (3.58) \\ \dfrac{dx_2}{dt} = x_3 & (3.59) \\ \dfrac{dx_3}{dt} = -a_2x_3 - a_1x_2 - a_0x_1 + u & (3.60) \end{cases}$$

Aceste ecuații pot fi scrise sub următoarea forma matriciala:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_2 & -a_1 & -a_0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot u \tag{3.61}$$

Procedând asemănător cu (3.52) se obține:

$$y = \begin{bmatrix} c_0 & c_1 & c_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.62}$$

Sub forma prescurtata ecuațiile de stare (3.61) şi (3.62) se pot scrie in modul următor:

$$\begin{cases} \dot{\underline{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{b} \cdot u(t) & (3.63) \\ y(t) = \underline{c}^T \cdot \underline{x}(t) & (3.64) \end{cases}$$

unde

$\dot{\underline{x}}, \underline{x}, \underline{b}, \underline{c}$ *sunt vectori*

$\underline{\underline{A}}$ *este matrice*

u *este scalar.*

In cazul mai multor mărimi de intrare şi ieşire ecuațiile de stare au următoarea forma generala:

$$\begin{cases} \dot{\underline{x}}(t) = \underline{\underline{A}} \cdot \underline{x}(t) + \underline{\underline{B}} \cdot \underline{u}(t) & (3.65) \\ \underline{y}(t) = \underline{\underline{C}} \cdot \underline{x}(t) + \underline{\underline{D}} \cdot \underline{u}(t) & (3.66) \end{cases}$$

Cu ajutorul ecuațiilor de stare se poate determina răspunsul sistemului $\underline{y}(t)$ atunci când se cunosc intrarea sa $\underline{u}(t)$ şi condițiile inițiale $\underline{x}(0)$. Sa consideram cazul mai simplu al unui sistem cu o singura intrare u(t), o singura ieşire y(t) şi o singura stare x(t). Relația (3.65) se transforma intr-o ecuație diferențiala de ordinul întâi:

$$\dot{x}(t) = a \cdot x(t) + b \cdot u(t) \qquad (3.67)$$

iar (3.66) se transforma intr-o ecuație algebrica de ordinul întâi:

$$y(t) = c \cdot x(t) + d \cdot u(t) \qquad (3.68)$$

Relația (3.67) poate modela, de exemplu, variația nivelului x(t) intr-un rezervor. Acesta depinde atât de debitul de intrare u(t), cât şi de nivelul inițial din rezervor x(0). Dar nivelul inițial poate fi echivalat cu o intrare (sursă) suplimentară. Sistemul fiind liniar se poate aplica principiul superpoziției efectelor provocate de fiecare cauza separat:

$$x(t) = x_n(t) + x_f(t) \qquad (3.69)$$

in care:

$x_n(t)$ este răspunsul natural al sistemului provocat numai de energia acumulata in elementele sistemului, deci pentru u(t)=0 (soluția generala a ecuației omogene, in termeni matematici);

$x_f(t)$ este răspunsul forțat al sistemului provocat numai de intrarea u(t) cu elementele sistemului relaxate, adică fără energii acumulate şi deci cu condiții inițiale nule x(0)=0 (soluția particulara a ecuației neomogene, in termeni matematici).

Din (3.67) rezulta ca răspunsul natural este soluția ecuației :

$$\dot{x}(t) = a \cdot x(t) \tag{3.70}$$

Se poate arata ca soluția cea mai generala este in acest caz:

$$x_n(t) = e^{at} \cdot K \tag{3.71}$$

in care K este o constanta care depinde de energia acumulata la momentul de timp inițial $t=t_0$ (condiția inițiala).

Dacă u(t)≠0 vom căuta o soluție generală a ecuației (3.67) de forma:

$$x(t) = e^{at} \cdot K(t) \tag{3.72}$$

in care K(t) este de data aceasta o funcție de timp.

Înlocuind soluția (3.72) in ecuația (3.67) rezulta după derivarea lui x(t):

$$\dot{x}(t) = a \cdot e^{at} \cdot K(t) + e^{at} \cdot \dot{K}(t) = a \cdot e^{at} \cdot K(t) + b \cdot u(t) \tag{3.73}$$

sau

$$\dot{K}(t) = e^{-at} \cdot b \cdot u(t) \tag{3.74}$$

Integrând ambele părți ale acestei relații rezulta:

$$K(t) = \int_{t_0}^{t} e^{-a\tau} \cdot b \cdot u(\tau) d\tau + K(t_0) \tag{3.75}$$

Înlocuind în (3.72) se obține:

$$x(t) = e^{at} \cdot \left[\int_{t_0}^{t} e^{-a\tau} \cdot b \cdot u(\tau) d\tau + K(t_0) \right] \tag{3.76}$$

Pentru determinarea lui $K(t_0)$ se face in (3.72) $t=t_0$:

$$K(t_0) = e^{-at_0} \cdot x(t_0) \tag{3.77}$$

şi deci soluția ecuației de stare (3.67) devine :

$$x(t) = e^{a(t-t_0)} \cdot x(t_0) + \int_{t_t}^{t} e^{a(t-\tau)} \cdot b \cdot u(\tau) d\tau \tag{3.78}$$

şi se poate distinge răspunsul natural $x_n(t)$ de cel forțat $x_f(t)$.

Prin analogie cu (3.67) se poate scrie soluția ecuației matriciale de stare (53):

$$\underline{x}(t) = e^{\underline{\underline{A}}(t-t_0)} \cdot \underline{x}(t_0) + \int_{t_0}^{t} e^{\underline{\underline{A}}(t-\tau)} \cdot \underline{\underline{B}} \cdot u(\tau) d\tau \tag{3.79}$$

in care exponențiala matriciala e^{At} se defineşte astfel:

$$\underline{\underline{\Phi}}(t) = e^{\underline{\underline{A}}t} = \underline{\underline{1}} + \underline{\underline{A}}t + \frac{1}{2!}(\underline{\underline{A}}t)^2 + \ldots + \frac{1}{n!}(\underline{\underline{A}}t)^n + \ldots \tag{3.80}$$

şi se numeşte matricea de tranziție a stărilor sistemului.

Răspunsul natural al sistemului rezulta imediat din (3.79) daca intrarea $\underline{u}(t) = 0$ şi se cunoaşte starea inițiala $\underline{x}(t_0)$:

$$\underline{x}(t) = e^{\underline{\underline{A}}(t-t_0)} \cdot \underline{x}(t_0) \tag{3.81}$$

Daca evaluarea variabilelor de stare se face la un interval constant de timp $T=t-t_0$ (70), atunci se poate calcula cu ajutorul (3.81):

$$\underline{x}(t_0 + T) = e^{\underline{\underline{A}}T} \cdot \underline{x}(t_0) \tag{3.82}$$

$$\underline{x}(t_0 + 2T) = e^{\underline{\underline{A}}T} \cdot \underline{x}(t_0 + T) \quad \ldots etc \tag{3.83}$$

in care *matricea de tranziție a stărilor* $e^{\underline{\underline{A}}t}$ *se calculează o singura data cu ajutorul seriei* (3.80).

Se observa ca

$$\frac{1}{10!} = 0.276 \cdot 10^{-6}$$

Deci daca se iau zece termeni in aproximarea seriei se obțin elementele matricei de transfer cu 10 cifre semnificative exacte. Câteva cazuri particulare sunt importante. Daca sistemul are o singura variabila de stare, atunci ecuația de stare şi răspunsul natural devin:

$$\dot{x}(t) = a \cdot x(t) \tag{3.84}$$

$$x_n(t) = e^{a(t-t_0)} \cdot x(t_0) \tag{3.85}$$

Daca t_0=0 şi condiția inițiala x(0)=1 atunci răspunsul natural este o exponențială:

$$x_n(t) = e^{at} \tag{3.86}$$

Pentru a=0 ecuația (3.84) devine

$$\dot{x}(t) = 0 \tag{3.87}$$

iar răspunsul natural este o treapta unitate

$$x(t) = u^1(t) \tag{3.88}$$

Folosind răspunsurile naturale ale unor sisteme particulare se poate transforma problema determinării *răspunsului general* (3.69) al unui sistem in problema determinării *răspunsului natural* al unui sistem lărgit.

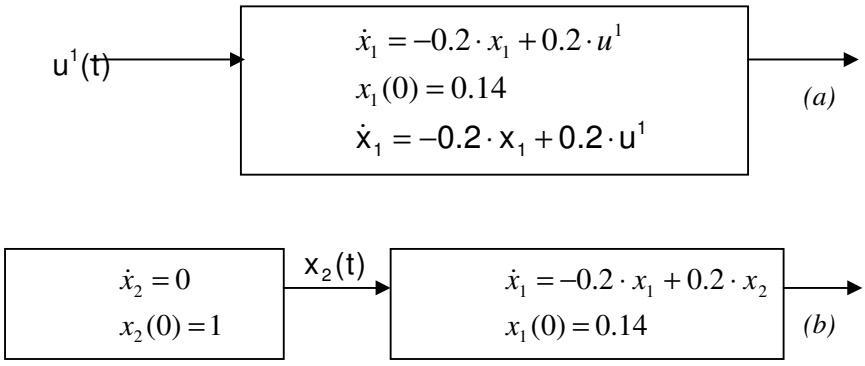


Fig. 3.12 Transformarea unui răspuns forțat in răspunsul natural al unui sistem lărgit

In *Fig. 3.12* se prezintă un exemplu. Sistemul este descris de o singura ecuație de stare de tipul (3.67) cu a=-0.2, b=0.2 şi condiția inițiala $x_1(0)$=0.14.Marimea de intrare este treapta unitate $u^1(t)$ care poate fi *generata* de ecuația (3.87).Introducând generatorul treptei unitate la intrarea in sistem (*Fig. 3.12* b) se obține un sistem mai mare, descris de doua ecuații de stare, dar *omogen*, fără intrare exogena (din afara sistemului):

$$\dot{x}_1 = -0.2 \cdot x_1 + 0.2 \cdot x_2 \qquad (3.89)$$

$$\dot{x}_1(0) = 0{,}14$$

$$\dot{x}_2 = 0 \qquad (3.90)$$

$$x_2(0) = 1$$

sau in notație matriciala:

$$\underline{\dot{x}} = \underline{\underline{A}} \cdot \underline{x}(t) \qquad (3.91)$$

in care:

$$\underline{\underline{A}} = \begin{bmatrix} -0.2 & 0.2 \\ 0 & 0 \end{bmatrix} \qquad (3.92)$$

$$\underline{x}(0) = \begin{bmatrix} 0.14 \\ 1 \end{bmatrix}$$

Alegem momentul inițial de timp t_0=0 şi intervalul de evaluare a variabilelor T=0.5 s. Soluția ecuațiilor de stare poate fi evaluata cu ajutorul relațiilor (3.82) şi (3.83).

$$\underline{\underline{AT}} = \begin{bmatrix} -0.2 & 0.2 \\ 0 & 0 \end{bmatrix} \cdot 0.5 = \begin{bmatrix} -0.1 & 0.1 \\ 0 & 0 \end{bmatrix}, \ (\underline{\underline{AT}})^2 = \begin{bmatrix} 0.01 & -0.01 \\ 0 & 0 \end{bmatrix}, \qquad (3.93)$$

$$(\underline{\underline{AT}})^3 = \begin{bmatrix} -0.001 & 0.001 \\ 0 & 0 \end{bmatrix} \qquad (3.94)$$

$$e^{\underline{\underline{AT}}} \cong \underline{\underline{1}} + (\underline{\underline{AT}}) + \frac{1}{2!}(\underline{\underline{AT}})^2 + \frac{1}{3!}(\underline{\underline{AT}})^3 =$$

$$= \begin{bmatrix} 1 - 0.1 + 0.005 - 0.00017 & 0.1 - 0.005 + 0.00017 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.90483 & 0.09517 \\ 0 & 1 \end{bmatrix} \qquad (3.95)$$

Aplicând (3.82) rezulta:

$$\underline{x}(0.5) = \begin{bmatrix} 0.90483 & 0.09517 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.14 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.22185 \\ 1 \end{bmatrix} \qquad (3.96)$$

Soluția exacta a ecuației de stare din *Fig. 3.12* a este următoarea:

$$x_1(t) = x_n(t) + x_f(t) = 0.14 \cdot e^{-\frac{t}{5}} + (1 - e^{-\frac{t}{5}}) \qquad (3.97)$$

Pentru t=0.5 s avem:

$$x_1(0.5) = 0.22184 \qquad (3.98)$$

adică aceeaşi cu soluția (3.96), cu excepția celei de-a 5-a cifre semnificative.

Cele patru modele – matematic, structural, funcțional şi cu variabile de stare – ale sistemului fizic pot fi folosite separat sau împreună. Alegerea soluției depinde foarte mult de tipul problemei, de experiența inginerească existentă şi de programele disponibile. Câteva medii de dezvoltare gratuite ale aplicațiilor pentru sisteme fizice sunt Dynast, KitSAS Matlab şi Scilab.

3.8. Modele în domeniul frecvenţă ale sistemului fizic

În domeniul frecvenţă datele[39] pot fi analizate, sintetizate, filtrate sau trecute prin diferite ferestre. Aceste prelucrări se realizează cu ajutorul seriei Fourier şi transformărilor Fourier, Laplace şi Z.

Prelucrare în domeniul frecvenţă se aplică în primul rând datelor care se modifică în timp dar variabila independentă poate fi şi alta, de exemplu spaţiul. Metoda a fost fundamentată la început pentru sistemele fizice *liniare* şi se bazează pe proprietatea remarcabilă a acestora de *fidelitate sinusoidală*, adică pe faptul că răspunsul unui sistem liniar la un semnal sinusoidal este tot un semnal sinusoidal, cu aceiaşi frecvenţă dar cu amplitudine şi fază diferite.

În afară de studiul propagării semnalelor prin sisteme liniare, prelucrarea în domeniul frecvenţă se mai foloseşte la identificarea surselor de semnal, analiza şi sinteza sunetelor şi imaginilor şi depistarea defecţiunilor.

Aplicarea largă a acestor metode în practică a condus la fabricarea în serie a unor microprocesoare specializate[40] pentru acest gen de operaţii care sunt folosite într-un evantai larg de aplicaţii, de la telefoane mobile la instrumente ştiinţifice.

Iată alte motive pentru care prelucrarea semnalelor în domeniul frecvenţă este importantă.

- Mişcarea naturală a sistemelor fizice provocată de energia acumulată este de formă sinusoidală sau exponenţială.
- Sistemele fizice pot fi caracterizate prin frecvenţele proprii.
- Prelucrarea numerică a semnalelor implică o analiză în domeniul frecvenţă pentru determinarea unei eşantionări corecte.
- Multe perturbaţii şi zgomote pot fi caracterizate prin frecvenţe ridicate şi pot fi filtrate.
- Analiza semnalelor aleatoare se face în domeniul frecvenţă.
- Descompunerea semnalelor periodice în serie Fourier este ortogonală. Prin urmare determinarea unor noi componente nu necesită recalcularea componentelor vechi.
- Există algoritmul de transformare Fourier rapidă (FFT).
- Multe semnale din natură sunt sinusoidale, de exemplu curentul alternativ, vibraţiile şi sunetele.
- Analiza stabilităţii sistemelor fizice şi stabilitatea algoritmilor de prelucrare numerică se face în domeniul frecvenţă.
- Analiza în domeniul frecvenţă este mai uşoară pentru calculul manual. Chiar dacă analiza se face asistată de calculator, interfaţa cu utilizatorul foloseşte tot conceptele din domeniul frecvenţă.
- Multe performanţe şi limitări fizice sunt exprimate în domeniul frecvenţă. De exemplu realizabilitatea fizică a sistemului impune ca

[39] Semnale, imagini, sunete.
[40] DSP – Digital Signal Processor.

funcţia sa de transfer să aibă gradul numărătorului mai mare sau egal cu gradul numitorului.

3.8.1. Scurt istoric

Analiza Fourier a fost preconizată pentru prima dată de Jean Baptiste Joseph Fourier (1768 – 1830), un matematician şi fizician francez. Fourier era interesat de propagarea căldurii şi a prezentat la Academia Franceză în 1807 un articol în care folosea sinusoidele la reprezentarea distribuţiilor de temperatură. Lucrarea afirma că orice semnal periodic poate fi reprezentat printr-o sumă de sinusoide alese corespunzător. Printre referenţii lucrării se aflau şi doi matematicieni faimoşi: Josepf Louis Lagrange (1736 – 1813) şi Pierre Simon de Laplace (1749 – 1827).

Laplace şi alţi referenţi au votat pentru publicarea lucrării dar Lagrange a protestat vehement. În următorii ani Lagrange a insistat că o astfel de abordare nu poate fi folosită pentru reprezentarea semnalelor cu colţuri, adică cu forme discontinue aşa cum sunt formele de undă rectangulare. Academia Franceză s-a înclinat în faţa prestigiului Lui Lagrange şi a respins lucrarea. Abia după 15 ani, după moartea lui Lagrange lucrarea a putut fi publicată. Din fericire Fourier a avut alte activităţi care i-au reţinut atenţia în această perioadă, politica, expediţia lui Napoleon În Egipt şi evitarea ghilotinei din timpul Revoluţiei Franceze.

Lagrange a avut dreptate când a afirmat că prin însumarea unor sinusoide nu se pot sintetiza semnale cu colţuri. Totuşi se pot obţine aproximări foarte bune. Eroarea care apare la analiza semnalelor datorită colţurilor este cunoscută în prezent sub numele de efect Gibbs.

3.8.2. Semnale sinusoidale

Există multe reprezentări ale semnalelor sinusoidale. O formă importantă este următoarea:

$$x(t) = A_k \cos(\omega_k t + \varphi_k) \quad , \quad -\infty < t < \infty \tag{3.99}$$

În această relaţie A reprezintă amplitudinea, ω pulsaţia şi φ faza.

O observaţie importantă o constituie faptul că sinusoidele sunt reprezentate de fapt prin cosinusoide şi acestea sunt definite de la minus infinit la plus infinit, diferind de reprezentarea tradiţională în care sinusoida începe din zero cu valoarea zero.

În relaţia (3.100) pulsaţia poate apare în următoarele două forme diferite

$$\omega_k = 2\pi f_k = \frac{2\pi}{T_k} \tag{3.100}$$

în care f este frecvenţa sinusoidei iar T este perioada sa.

Un rol important în prelucrarea datelor în domeniul frecvenţă îl joacă relaţia lui Euler:

$$e^{j\beta} = \cos\beta + j\sin\beta \tag{3.101}$$

Pentru unghiuri negative,

$$e^{-j\beta} = \cos\beta - j\sin\beta \tag{3.102}$$

Adunând şi scăzând aceste relații rezultă

$$\cos\beta = \frac{e^{j\beta} + e^{-j\beta}}{2} \tag{3.103}$$

$$\sin\beta = \frac{e^{j\beta} - e^{-j\beta}}{2j} \tag{3.104}$$

Înlocuind pe (3.103) în (3.99)) se obține forma complexă a semnalului sinusoidal de pulsație ω_k:

$$x(t) = A_k \frac{e^{j(\omega_k t+\varphi_k)} + e^{-j(\omega_k t+\varphi_k)}}{2} = X_k e^{j\omega_k t} + X_{-k} e^{-j\omega_k t} \tag{3.105}$$

în care

$$X_k = \frac{A_k}{2} e^{j\varphi_k} \tag{3.106}$$

$$X_{-k} = \frac{A_k}{2} e^{-j\varphi_k} \tag{3.107}$$

- ***Propagarea semnalelor sinusoidale prin sisteme liniare***

Forma complexă (3.105) este potrivită pentru studiul propagării sinusoidelor prin sistemele liniare. Aplicând principiul superpoziției se poate studia propagarea celor două componente separat. Să considerăm prima componentă numită sinusoidă complexă:

$$x_1(t) = X_k e^{j\omega_k t} \tag{3.108}$$

Datorită fidelității sinusoidale ieşirea sistemului liniar va avea aceiaşi pulsație ω_k dar o amplitudine şi fază diferite

$$y_1(t) = Y_k e^{j\omega_k t} \tag{3.109}$$

Amplitudinea semnalului de ieşire este un număr complex de forma:

$$Y_k = \frac{B_k}{2} e^{j\psi_k} \tag{3.110}$$

Se procedează la fel şi cu a doua componentă $x_2(t)$ şi se obține $y_2(t)$. Răspunsul sistemului va fi:

$$y(t) = y_1(t) + y_2(t) \qquad (3.111)$$

Sistemele fizice se comportă la trecerea semnalelor prin ele ca nişte filtre trece jos, adică atenuează puternic sinusoidele cu frecvențe înalte.

- ***Filtrul analogic de ordin unu. Funcția de transfer în domeniul frecvență***

Se analizează efectul de filtru trece jos a sistemului pompă – rezervor. Acesta lasă să treacă prin el numai semnalele sinusoidale de frecvență joasă. La început se obține modelul matematic liniar.

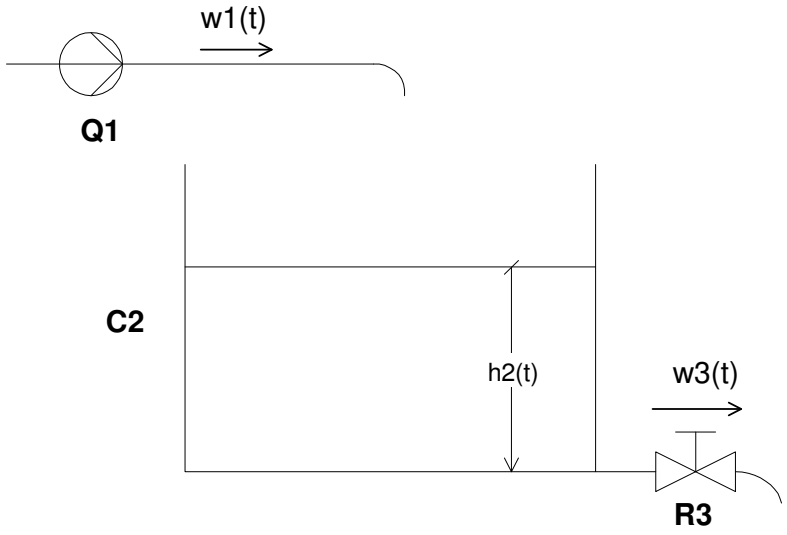


Fig. 3.13 Sistemul pompă rezervor

Aplicând legea conservării masei[41] la sistemul pompă - rezervor din Fig. 3.13 rezultă următoarea ecuație diferențială neliniară:

[41] Masa acumulată în rezervor este egală cu diferența dintre debitul masic de intrare şi debitul masic de ieşire.

$$\frac{d}{dt}(A\rho h_2) = w_1 - w_3 = w_1 - \alpha\sqrt{h_2} \qquad (3.112)$$

sau

$$\frac{dh_2}{dt} + c\sqrt{h_2} = bw_1 \qquad (3.113)$$

Considerând variații mici ale nivelului Δh_2 în jurul nivelului h_0 se obține:

$$\Delta(\frac{dh_2}{dt}) + c\Delta(\sqrt{h_2}) = b\Delta w_1 \qquad (3.114)$$

Pentru variații foarte mici operatorul Δ poate fi înlocuit cu operatorul diferențial d.

$$d(\sqrt{h_2}) = \left.\frac{d\sqrt{h_2}}{dh_2}\right|_{h_2=h_0} dh_2 = \frac{1}{2\sqrt{h_0}} dh_2 \qquad (3.115)$$

Pentru simplificarea scrierii notăm $dh_2 = h$ şi $dw_1 = w$. Ecuația diferențială, de data această liniară, este:

$$\frac{dh}{dt} + \frac{c}{2\sqrt{h_0}} h = b.w \qquad (3.116)$$

Această ecuație este valabilă numai pentru variații mici ale nivelului h în jurul valorii inițiale h_0.

Modelul liniar (3.116) al sistemului se scrie sub forma:

$$T\frac{dy(t)}{dt} + y(t) = Kx(t) \qquad (3.117)$$

în care T este o constantă de timp măsurată în secunde iar K o constantă de proporționalitate.

Să considerăm exemplul numeric cu c=2 şi b=1. Se calculează semnalul de ieşire al sistemului, adică nivelul, *în regim staționar* pentru o intrare constantă. Un semnal constant de la minus infinit la plus infinit poate fi considerat că provine dintr-o sinusoidă (3.99) cu pulsația egală cu zero[42]. Se consideră următoarele situații:

1. Rezervorul din Fig. 3.13 este gol, debitul constant w_1=12, modelul este neliniar. Din (3.113) rezultă h_{2st}=36.
2. Rezervorul umplut la nivelul h_0=36, debitul este constant şi mic w=1,2 iar modelul este liniar. Folosind relația (3.116) se obține h_{st}=7,2. Deci

[42] În acest caz şi frecvența este zero iar perioada are o valoare infinită.

valoarea staționară a nivelului din rezervorul din Fig. 3.13 va fi h_{2st}=43,2.

3. Daca încercăm să rezolvăm cazul precedent fără să folosim modelul liniarizat prin aproximare (3.116) al rezervorului vom introduce în modelul neliniar(3.113) valoarea debitului total w_1=12+1,2=13,2 şi rezultă h_{2st}=43,56. Rezultatul este foarte apropiat de cel obținut prin liniarizarea sistemului de la punctul precedent.

Să determinăm acum răspunsul sistemului liniar (3.117) pompă – rezervor la un debit sinusoidal staționar de forma (3.99). Descompunem sinusoida în două componente complexe conform relației (3.105). Deoarece sistemul este liniar şi se aplică principiul superpoziției, calculăm la început răspunsul la prima componentă (3.108). Răspunsul sistemului este tot sinusoida şi are forma (3.109). Înlocuind în (3.117) se obține:

$$Tj\omega_k Y_k e^{j\omega_k t} + Y_k e^{j\omega_k t} = KX_k e^{j\omega_k t} \qquad (3.118)$$

Aceasta este o ecuație algebrică liniară şi putem calcula ieşirea Y_k în funcție de intrarea în sistem X_k.

$$Y_k = \frac{K}{1+Tj\omega_k} X_k = G_k X_k \qquad (3.119)$$

în care

$$G_k = \frac{Y_k}{X_k} = \frac{K}{1+Tj\omega_k} \qquad (3.120)$$

G_k este *constanta de transfer* a sistemului liniar pentru intrarea sinusoidală cu pulsația ω_k. Această constantă este un număr complex a cărui valoare depinde de valoarea pulsației ω_k a semnalului sinusoidal care se propagă prin sistem.

Dacă se consideră în (3.120) toate valorile posibile ale pulsației $\omega=\omega_k$ se defineşte *funcția de transfer în domeniul frecvență G(jω)* a sistemului la propagarea semnalului sinusoidal de la intrarea sa la ieşirea lui în regim staționar:

$$G(j\omega) = \frac{K}{1+Tj\omega} = \left|G(j\omega)\right| e^{j\theta} \qquad (3.121)$$

Modulul funcției de transfer este

$$\left|G(j\omega)\right| = \frac{K}{\sqrt{1+T^2\omega^2}} \qquad (3.122)$$

Argumentul funcției de transfer este:

$$\theta = \arg G(j\omega) = -\arctan(\omega T) \qquad (3.123)$$

Pentru a obține constanta de transfer a unui semnal sinusoidal cu pulsația ω_k se înlocuieşte această valoare în (3.121), (3.122) şi (3.123)

Folosind relațiile (3.106), (3.110),(3.119) şi (3.121) se poate determina valoarea semnalului X_k după propagarea lui prin sistem:

$$B_k = |G(j\omega_k)|A_k \tag{3.124}$$

$$\psi_k = \theta_k + \varphi_k \tag{3.125}$$

Datorită fidelității sinusoidale semnalul după propagarea prin sistem va fi de forma:

$$y(t) = B_k \cos(\omega_k t + \psi_k) \tag{3.126}$$

Se observă că ieşirea sinusoidală este complet determinată de (3.124) şi (3.125) dacă se cunoaşte modulul şi argumentul funcției de transfer pentru pulsația ω_k. Nu mai este necesar calculul răspunsului sistemului la a doua componentă X_{-k} a sinusoidei şi aplicarea, după aceia, a principiului superpoziției.

Să considerăm un exemplu numeric. Adoptăm aceleaşi valori ca şi în cazul precedent al unei intrări constante. Deci c=2, b=1, y_0=36. Din (3.116) şi (3.117) rezultă T=6 h şi K=1/6.

Funcția de transfer a sistemului este:

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \frac{6}{1+6j\omega} \tag{3.127}$$

Pentru ca modelul liniar să funcționeze este necesar ca semnalul de intrare să aibă valori mici în jurul punctului static de funcționare. Ca şi în cazul precedent alegem o sinusoida cu amplitudinea mică:

$$x(t) = 1{,}2\cos(\frac{1}{4}t - \frac{\pi}{2}) \tag{3.128}$$

Pulsația sinusoidei este ω_k=0,25. Înlocuind în (3.122), (3.123) şi folosind (3.124), (3.125) se obține valoarea sinusoidei la ieşirea din sistem

$$|G(j\omega_k)| = \frac{K}{\sqrt{1+T^2\omega_k^2}} = \frac{6}{\sqrt{1+6^2\cdot 0{,}25^2}} = 3{,}32 \tag{3.129}$$

$$\theta_k = \arctan(-\omega_k T) = -\arctan(0{,}25\cdot 6) = -0{,}98\ radiani \tag{3.130}$$

$$B_k = |G(j\omega_k)|A_k = 3{,}3\cdot 1{,}2 = 4 \tag{3.131}$$

Faza sinusoidei la ieşirea din sistem este

$$\psi_k = \theta_k + \varphi_k = -0{,}98 - \frac{\pi}{2} = -4{,}12 \text{ radiani} \tag{3.132}$$

Sinusoida după propagarea prin sistem va fi:

$$y(t) = 4\cos(\frac{1}{4}t - 4{,}12) \tag{3.133}$$

Pentru o sinusoidă cu pulsaţia ω_k=0 răspunsul este 6.1,2=7,2. Se observă că sistemul amplifică sinusoida cu pulsaţia ω_k=0,25 de aproape două ori mai puţin decât sinusoida cu pulsaţia zero şi deci acţionează ca un filtru trece jos. Întârzierea ieşirii sinusoidale faţă de intrarea sinusoidală este egală cu $\theta_k T/2\pi$=0,93 h.

- ***Filtrul analogic de ordin doi***

In Fig. 3.14 se prezintă schematic un motor electric aşezat pe o fundaţie. Comportarea acestui sistem este echivalentă cu cea a unui filtru analogic trece jos de ordin doi. Acest filtru lasă să treacă prin sistem numai semnalele sinusoidale cu frecvenţa joase.

Echilibrul forţelor conduce la următorul model matematic pentru acest sistem:

$$x(t) = m\frac{d^2y(t)}{dt^2} + c\frac{dy(t)}{dt} + ky(t) \tag{3.134}$$

Variaţia în timp a forţei este sinusoidală de forma (3.99. Datorită faptului că sistemul este sinusoidal se bucură de proprietatea de fidelitate sinusoidală şi ieşirea sistemului, deplasarea fundaţiei este tot sinusoidală. Aplicând aceiaşi metodă ca şi în paragraful precedent rezultă următoarea funcţie de transfer în domeniul frecvenţă pentru acest sistem:

$$G(j\omega) = \frac{1}{m(j\omega)^2 + c(j\omega) + k} = K\frac{\omega_n^2}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2} \tag{3.135}$$

în care ω_n este pulsaţia naturală iar ζ este fracţiunea de amortizare critică.

Această funcție de transfer de ordinul doi este în acelaşi timp şi modelul matematic standard cu ajutorul căruia se definesc performanțele unui sistem de reglare automată.

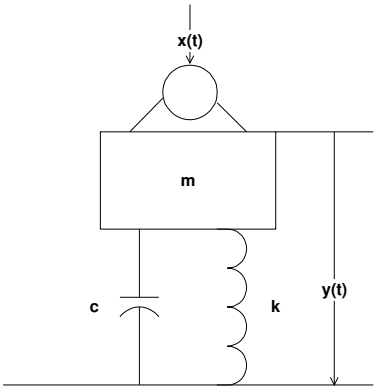


Fig. 3.14. Oscilatorul mecanic.

Programul de simulare pentru un sistem descris de un model matematic tip ecuație diferențială de ordinul doi cu coeficienți constanți este prezentat în Fig. 3.15. Graficul modulului şi fazei funcției de transfer în domeniul frecvență apare în Fig. 3.16.

```
//filtru2poli.sci
clear
s=poly(0,'s');                  //definirea variabilei s folosite in polinom
g=1/(1+0.44*s+s^2);             //definirea raport doua polinoame
G=syslin('c',g);                //definirea functiei de transfer
[f,repf]=repfreq( G,0.01,0.5); //calcul raspuns in frecventa, 100 frecvente intre 0.01 si 0.1
xbasc();
[fi,db]=phasemag(repf,'m');    //calculul fazei si modulului
db=100*(10^(db/20));              //trecerea din decibeli in amplitudine amplificata de 10 ori
plot2d([f',f'],[fi',db']);          //tiparire doua grafice
xgrid();
xtitle('Analiza in domeniul frecventa - filtrul cu doi poli', 'f[Hz]',...
'100*Modul, Faza[grade]');
```

Fig. 3.15 Programul Scilab pentru calculul modulului şi fazei funcției de transfer

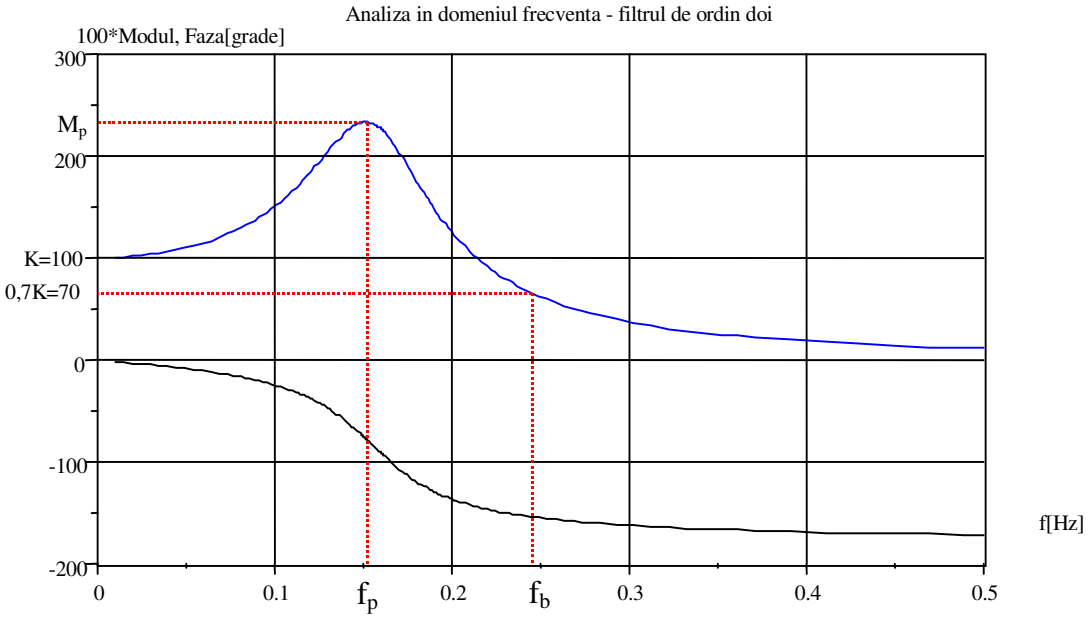


Fig. 3.16 Graficul modulului şi fazei funcției de transfer

În Fig. 3.16 există câteva puncte de referință de referință. Conform relației (3.135) pulsației ω=0 îi corespunde valoarea K. Prin definiție punctul corespunzător valorii 0,707K definește banda de trecere a filtrului caracterizată de frecvența f_p. Amplificarea maximă a filtrului este M_p şi apare la pulsația ω_p. Relațiile dintre aceste mărimi şi fracțiunea de amortizare critică ζ şi pulsația naturală ω_n sunt prezentate în continuare.

$$M_p = \frac{1}{2\zeta\sqrt{(1-\zeta^2)}} \qquad (3.136)$$

$$\omega_p = \omega_n\sqrt{(1-2\zeta^2)} \qquad (3.137)$$

Din motive practice graficul din Fig. 3.16 este prezentat în coordonate logaritmice. Pe ordonată se foloseşte unitatea de măsură decibelul. Modulul exprimat în decibeli se obține cu ajutorul următoarei relații:

$$M_{dB} = 20\log_{10}(M) \qquad (3.138)$$

3.8.3. Semnale periodice. Seria Fourier

Semnalele periodice cu perioada T_0 pot fi exprimate ca o sumă de sinusoide cu ajutorul seriei Fourier.

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{k=\infty}(a_k \cos k\omega_0 t + b_k \sin k\omega_0 t)$$

(3.139)

$$a_0 = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} x(t)dt \qquad (3.140)$$

$$a_k = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} x(t)\cos(k\omega_0 t)dt \qquad (3.141)$$

$$b_k = \frac{2}{T_0} \int_{t_0}^{t_0+T_0} x(t)\sin(k\omega_0 t)dt \qquad (3.142)$$

Folosind relația lui Euler se poate obține seria Fourier sub formă complexă:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k e^{jk\omega_0 t} \qquad (3.143)$$

$$X_k = \frac{1}{T_0} \int_{t_0}^{t_0+T_0} x(t)e^{-jk\omega_0 t} \qquad (3.144)$$

Semnalul periodic $x_P(t)$ din *Fig. 3.17*a) a fost obținut prin repetarea periodică, cu perioada T_0, a unui impuls x(t) cu forma unui dinte de ferestrău. Modulul componentelor X_k este prezentat în *Fig. 3.17*b).

3.8.4. Semnale impuls. Transformarea Fourier. Transformarea Laplace

Dacă perioada T_0 de repetiție a dintelui de ferestrău crește spre infinit semnalul periodic se transformă într-un impuls iar liniile spectrului din *Fig. 3.17*b) se apropie formând o funcție continuă. Suma din (3.143) se transformă în integrală şi se obține transformarea Fourier determinată prin relațiile următoare:

$$X(j\omega) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad (3.145)$$

$$x(t) = F^{-1}\{X(j\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega)e^{j\omega t}d\omega \qquad (3.146)$$

În *Fig.*

3.17 se observă că periodizarea în domeniul timp duce la o eşantionare în domeniul frecvenţă. Reciproca este de asemenea valabilă. O eşantionare în domeniul timp conduce la o periodizare în domeniul frecvenţă. Dacă spectrul continuu este infinit, periodizarea sa conduce la o suprapunere a valorilor provocând fenomenul de aliere sau suprapunere. Dacă din acest spectru cu suprapuneri se încearcă obţinerea semnalului iniţial se obţin rezultate eronate. Evitarea acestei erori se face impunând condiţia ca frecvenţa de eşantionare să fie mai mare decât de două ori frecvenţa maximă din spectrul continuu.

$$f_e = 2 f_{\max} \tag{3.147}$$

Relaţia (3.147) este numită teorema eşantionării, teorema lui Shannon sau teorema lui Nyquist. Frecvenţa $f_e/2$ este numită adeseori frecvenţa Nyquist.

În *Fig. 3.17*c) frecvenţa de eşantionare este $f_e=1/T$. În domeniul frecvenţă din *Fig. 3.17*d) această frecvenţă, sau pulsaţia corespunzătoare Ω, măsoară perioada de repetiţie a spectrului. Se observă că Ω este de două ori mai mare decât pulsaţia maximă a spectrului

Din relaţiile (3.144) şi (3.145) rezultă:

$$X(j\omega) = \lim_{T_0 \to \infty} \frac{X_k}{f} \tag{3.148}$$

Datorită acestei legături între spectrul X_k şi transformarea Fourier, aceasta din urmă se mai numeşte şi *densitate spectrală*.

Integrala (3.145) nu este convergentă pentru unele semnale frecvent întâlnite în practică, de exemplu treapta unitară. Acest dificultate este depăşită dacă se înlocuieşte jω cu frecvenţa complexă s definită astfel:

$$s = \sigma + j\omega \tag{3.149}$$

Relaţia (3.145) se transformă în transformata Laplace bilaterală:

$$X(s) = L\{x(t)\} = \int_{-\infty}^{\infty} x(t) e^{-st}\, dt \tag{3.150}$$

Transformarea Laplace inversă va fi:

$$x(t) = L^{-1}\{X(s)\} = \frac{1}{2\pi j} \int_{s=\sigma_0 - j\infty}^{\sigma_0 + j\infty} X(s) e^{st}\, ds \tag{3.151}$$

Multe semnale x(t) au însă un început, adică un moment înaintea căruia sunt nule. Considerând originea timpului în acel moment se obţine transformata Laplace unilaterală:

$$X(s) = L\{x(t)\} = \int_0^\infty x(t)e^{-st}dt \qquad (3.152)$$

Transformarea Laplace inversă (3.151) rămâne aceiaşi şi pentru transformarea Laplace unilaterală. Diferența dintre ele apare atunci când sistemul prin care se propagă semnalul are condiții inițiale diferite de zero. Pentru condiții inițiale nule cele două transformări sunt identice.

Funcția de transfer a unui filtru pentru semnale impuls se obține din funcția de transfer la semnale sinusoidale prin înlocuirea lui jω cu s. De exemplu, funcția de transfer (3.135)a filtrului de ordinul doi devine:

$$G(s) = \frac{Y(s)}{X(s)} = K\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (3.153)$$

Funcția de transfer se defineşte, de obicei, sub forma cea mai generală prin raportul dintre transformata Laplace a semnalului de ieşire şi a transformatei Laplace a semnalului de intrare în condiții inițiale nule. Înlocuind pe s cu jω se obține funcția de transfer definită cu ajutorul transformatei Fourier prin raportul densităților spectrale ale semnalelor de ieşire şi intrare.

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = K\frac{\omega_n^2}{(j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2} \qquad (3.154)$$

Dacă semnalele de intrare şi ieşire sunt sinusoidale atunci funcția de transfer se obține din forma generală (3.153) înlocuind pe s cu $j\omega_k$ şi se obține:

$$G_k = \frac{Y_k}{X_k} = K\frac{\omega_n^2}{(j\omega_k)^2 + 2\zeta\omega_n(j\omega_k) + \omega_n^2} \qquad (3.155)$$

De data aceasta funcția de transfer este definită prin raportul sinusoidelor complexe (3.108) de intrare şi ieşire.

În practica prelucrării datelor transformatele Fourier şi Laplace sunt calculate prin reducerea lor la serii Fourier. Pentru aceasta semnalele impuls sunt periodizarea şi eşantionate ca în *Fig. 3.17*. Se foloseşte algoritmul transformării Fourier rapide.

Trecerea semnalelor prin sisteme liniare, de exemplu filtre, provoacă modificarea lor prin efectuarea asupra lor a unor operații. Efectul acestor operații în domeniul timp şi în domeniul frecvență este prezentat în *Tab. 3.3*. Se observă că transformarea Laplace unilaterală ia în considerare în mod *explicit* existența energiilor acumulate în elementele sistemului şi caracterizate prin existența unor condiții inițiale diferite de zero. Transformarea Laplace bilaterală, ca şi transformarea Fourier care constituie un caz particular al ei, trebuie să considere separat semnalele datorate energiilor acumulate inițial. Dacă sistemul este relaxat, adică nu are energie acumulată în elemente şi deci condițiile inițiale sunt nule, atunci transformările Laplace bilaterală şi unilaterală sunt echivalente.

Tab. 3.3 Prelucrarea semnalelor în domeniul timp şi în domeniul frecvenţă

Operatia	*Domeniul timp x(t)*	*Domeniul frecventa*	
		Transformarea Laplace bilaterala X(s)=L{x(t)}	*Transformarea Laplace unilaterala X(s)=L{x(t)}*
Derivarea	$\frac{dx(t)}{dt}$	$sX(s)$	$sX(s)-x(0^-)$
Integrarea	$\int x(t)dt$	$\frac{1}{s}X(s)$	$\frac{1}{s}X(s)+\frac{1}{s}\left[\int x(t)dt\right]_{t=0^-}$
Insumarea	$k_1x_1(t)+k_2x_2(t)$	$k_1X_1(s)+k_2X_2(s)$	
Intarzierea	$x(t-\tau)$	$e^{-s\tau}X(s)$	
Convolutia	$\int_{-\infty}^{t} x(\tau)h(t-\tau)d\tau$	$H(s)X(s)$	
Schimbare scara	$X(\frac{t}{\lambda})$	$\lvert\lambda\rvert X(\lambda s)$	

În *Tab. 3.3* se observă că operaţiile de derivare şi integrare a semnalelor se transformă în domeniul frecvenţă în operaţii de înmulţire şi împărţire care sunt mult mai simplu de realizat.

În Tab. 3.4 sunt prezentate Transformatele Laplace ale semnalelor cel mai des întâlnite în practică. Se observă că acestea sunt înmulţite în domeniul timp cu funcţia treaptă unitara asigurând prin aceasta începerea lor la momentul de timp egal cu zero.

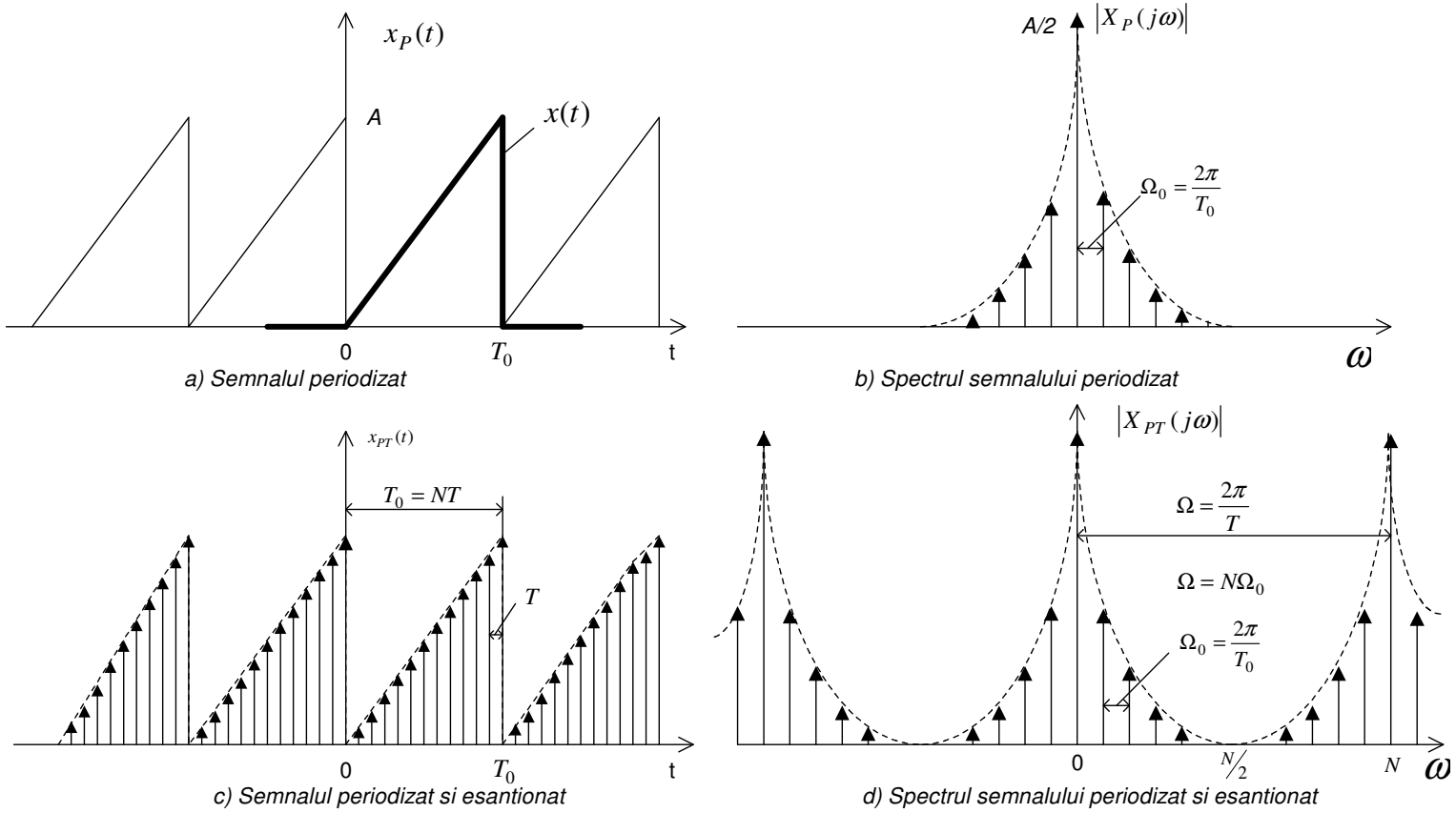
Tab. 3.4 Transformate Laplace ale semnalelor

$x(t)$	$X(s)$
$\delta(t)$	1
$u^1(t)$	$\frac{1}{s}$
$u^1(t)t^n$	$\frac{n!}{s^{n+1}}$
$u^1(t)e^{-\alpha t}$	$\frac{1}{(s+\alpha)}$
$u^1(t)e^{-\alpha t}\sin\beta t$	$\frac{\beta}{(s+\alpha)^2+\beta^2}$
$u^1(t)e^{-\alpha t}\cos\beta t$	$\frac{s+\alpha}{(s+\alpha)^2+\beta^2}$
$u^1(t)\left\{1-e^{-\frac{t}{T}}(1+\frac{t}{1!T}+\frac{t^2}{2!T^2}+\cdots+\frac{t^{n-1}}{(n-1)!T^{n-1}})\right\}$	$\frac{1}{(1+Ts)^n}\frac{1}{s}$
$u^1(t)\left\{1-\frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\cos\left[(\omega_n\sqrt{1-\zeta^2})t+arctg\frac{\zeta}{\sqrt{1-\zeta^2}}\right]\right\}$	$\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\frac{1}{s}$

În Tab. 3.5 sunt prezentate transformatele Fourier ale unor semnale frecvent întâlnite în practică. Pentru semnalele impuls care încep de la zero transformata Fourier este identică cu Transformata Laplace din Tab. 3.4 dacă se înlocuieşte s cu jω.

Tab. 3.5 Transformatele Fourier ale semnalelor

$x(t)$	$X(j\omega)$
$\delta(t)$	1
$u^1(t)$	$\pi\delta(\omega)+\frac{1}{j\omega}$
$u^1(t)e^{-\alpha t}$	$\frac{1}{j\omega+\alpha}$
$u^1(t)e^{-\alpha t}\sin\beta t$	$\frac{\beta}{(j\omega+\alpha)^2+\beta^2}$
$u^1(t)e^{-\alpha t}\cos\beta t$	$\frac{j\omega+\alpha}{(j\omega+\alpha)^2+\beta^2}$
$e^{j\alpha t}$	$2\pi\delta(\omega-\alpha)$
1	$2\pi\delta(\omega)$
$\cos\beta t$	$\pi[\delta(\omega-\beta)+\delta(\omega+\beta)]$
$\sin\beta t$	$\frac{\pi}{j}[\delta(\omega-\beta)-\delta(\omega+\beta)]$



a) Semnalul periodizat

b) Spectrul semnalului periodizat

c) Semnalul periodizat si esantionat

d) Spectrul semnalului periodizat si esantionat

Fig. 3.17 Efectul periodizării şi eşantionării semnalelor

3.8.5. Semnificaţii fizice în domeniul frecvenţă

Studiul în domeniul frecvenţă al semnalelor este folosit frecvent în practică datorită avantajelor sale dar are un caracter mai puţin intuitiv. Din această cauză legăturile directe între domeniul timp şi domeniul frecvenţă prezintă o importanţă deosebită

Variabilele independente în cele două domenii sunt timpul şi frecvenţa. În domeniul frecvenţă se foloseşte frecvent drept variabilă independentă şi pulsaţia ω.

$$\omega = 2\pi f = \frac{2\pi}{T}\left[rad/s\right] \qquad (3.156)$$

în care f este frecvenţa [Hz] iar T [s] este perioada sinusoidelor de intrare şi ieşire.

Transformatele Laplace sau Fourier ale semnalelor au dimensiunea semnalului original înmulţită cu timpul datorită relaţiilor de definiţie.

Între domeniul timp şi domeniul frecvenţă există următoarele relaţii directe.

Teorema valorii finale.

$$x(\infty) = \lim_{t\to\infty} x(t) = \lim_{s\to 0}\left[sX(s)\right] \qquad (3.157)$$

Teorema valorii iniţiale.

$$x(0) = \lim_{t\to 0} x(t) = \lim_{s\to\infty}\left[sX(s)\right] \qquad (3.158)$$

Teorema lui Parceval.

$$\int_{-\infty}^{\infty} x^2(t)dt = \frac{1}{2\pi}\int_{-\infty}^{\infty}\left|X(j\omega)\right|^2 d\omega \qquad (3.159)$$

în care $\left|X(j\omega)\right|$ este modulul transformării Fourier.

Integrala din stânga a relaţiei este proporţională cu energia semnalului. Din această cauză $\left|X(j\omega)\right|^2$, o mărime măsurabilă uşor, se numeşte densitate spectrală a energiei.

3.8.6. Răspunsul sistemelor la semnale impuls în domeniul frecvenţă

Pentru un sistem liniar, datorită aplicabilităţii principiului superpoziţiei, dacă se cunoaşte răspunsul forţat al sistemului la o intrare particulară atunci se poate calcula răspunsul forţat al sistemului la orice intrare. De exemplu dacă se cunoaşte funcţia pondere h(t), care este răspunsul forţat al sistemului la un impuls Dirac, integrala de convoluţie permite calculul răspunsului pentru o intrare oarecare.

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = h(t) * x(t) \qquad (3.160)$$

Aplicând transformata Laplace acestei relaţii se obţine conform Tab. 3.1:

$$Y(s) = L\{h(t) * x(t)\} = H(s)U(s) \tag{3.161}$$

în care H(s) este transformata Laplace a funcţiei pondere şi coincide cu funcţia de transfer definită în paragrafele precedente folosind semnale sinusoidale. Relaţii asemănătoare se pot definii şi pentru alte semnale tip de intrare, de exemplu funcţia treaptă unitară.

Funcţia de transfer se determină experimental sau analitic aplicând transformata Laplace ecuaţiilor diferenţiale care reprezintă modelul sistemului. Să prezentăm un exemplu.

Modelul (3.117) al sistemului pompă – rezervor poate avea următoarea realizare:

$$5\frac{dy(t)}{dt} + y(t) = x(t) \tag{3.162}$$

cu nivelul iniţial în rezervor

$$y(0) = 0{,}14 \tag{3.163}$$

Aplicând transformata Laplace ecuaţiei (3.163) rezultă:

$$sY(s) - y(0) = -0{,}2Y(s) + 0{,}2X(s) \tag{3.164}$$

Răspunsul la o intrare oarecare X(s) va fi:

$$Y(s) = \frac{0{,}2}{s+0{,}2}X(s) + \frac{y(0)}{s+0{,}2} \tag{3.165}$$

Pentru condiţii iniţiale nule rezultă funcţia de transfer:

$$H(s) = \frac{0{,}2}{s+0{,}2} = \frac{1}{5s+1} \tag{3.166}$$

După determinarea funcţiei de transfer pe cale experimentală sau analitică, cum a fost cazul aici, de determină din tabele, de exemplu Tab. 3.4, Transformarea Laplace X(s) pentru intrarea dorită x(t). Dacă ,se doreşte răspunsul la o treaptă unitară $x(t)=u^1(t)$ se obţine:

$$X(s) = \frac{1}{s} \tag{3.167}$$

Introducând această valoare şi condiţia iniţială (3.161) în (3.165) rezultă răspunsul sistemului la o treaptă de unitară de intrare care poartă denumirea de răspuns indicial.

$$Y(s) = \frac{0,2}{s+0,2}\frac{1}{s} + \frac{0,14}{s+0,2} = Y_f(s) + Y_n(s) \qquad (3.168)$$

în care $Y_f(s)$ este răspunsul forţat iar $Y_n(s)$ este răspunsul natural al sistemului. Prin aplicarea principiului superpoziţiei cele două componente ale răspunsului pot fi prelucrate separat. Trecerea din domeniul frecvenţă în domeniul timp se face tot cu ajutorul tabelelor. În Tab. 3.4 componentei naturale îi corespunde:

$$y_n(t) = 0,14u^1(t)e^{-0,2t} \qquad (3.169)$$

Componenta forţată $Y_f(s)$ nu are însă o corespondenţă directă în Tab. 3.4. Există însă tabele de transformate Laplace mai complete. O altă cale de rezolvare a problemei constă în folosirea Tab. 3.4 şi descompunând pe $Y_f(s)$ în fracţii simple.

$$Y_f(s) = \frac{0,2}{s+0,2}\frac{1}{s} = \frac{A}{s} + \frac{B}{s+0,2} \qquad (3.170)$$

Aducând la acelaşi numitor şi identificând coeficienţii variabilei s din (3.170) rezultă A=1 şi B=-1. Cu $Y_f(s)$ descompus se poate folosi Tab. 3.4 şi rezultă:

$$y_f(t) = u^1(t) - u^1(t)e^{-0,2t} = u^1(t)[1 - e^{-0,2t}] \qquad (3.171)$$

Suprapunând răspunsul forţat şi răspunsul natural se obţine:

$$y(t) = y_f(t) + y_n(t) = u^1(t)[1 - e^{-0,2t}] + 0,14u^1(t)e^{-0,2t} = u^1(t)[1 - 0,86e^{-0,2t}] \qquad (3.172)$$

4. Sisteme logice combinaţionale

Funcţionarea multor dispozitive şi echipamente tehnice poate fi descrisă aproximativ cu ajutorul unor variabile binare, numite *biţi,* care pot lua numai două valori: {acţionat, neacţionat}, {0, 1}, {oprit, pornit}, {stinsă, aprinsă}, etc. Cu aceste variabile se pot realiza diferite operaţii asemănătoare cu cele din logica propoziţională, algebra booleană binară sau algebra părţilor unei mulţimi, după cum se vede în Tab. 4.1. În lucrările consacrate sistemelor de conducere automată se foloseşte o terminologie şi o notaţie împrumutată din aceste algebre şi din logică. În plus se adaugă unii termeni folosiţi în tehnica circuitelor logice electronice integrate şi tehnica calculatoarelor electronice.

Tab. 4.1 Comparaţie între diferite algebre

Algebra părţilor unei mulţimi	***Algebra booleana binară***	***Logica propoziţională***
Partea ***A*** a mulţimii ***E***	x ϵ {0, 1}	Enunţ {fals, advarat}
Intersecţie (∩)	Înmulţire booleana (.)	Conector de coordonare (SI) notat (Λ)
Reuniune (U)	Adunare booleana (+)	Conector de coordonare (SAU) notat (V)
Complementare (¯)	Complementare (¯)	Conector de modificare (NU) sau negare notat (┐)

Modelul funcţional al unui sistem logic combinaţional simplu este prezentat în Fig. 4.1. Se disting intrările *a* şi *b,* ieşirea *y* şi relaţia logică dintre ele *f*.

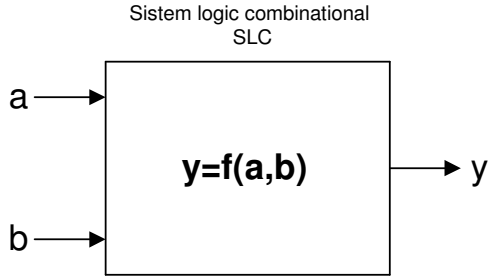


Fig. 4.1 Schema bloc a unui sistem logic combinaţional

Se disting următoarele probleme care trebuiesc rezolvate în legătură cu sistemele logice combinaţionale.

- ✓ ***Analiza***.

Se cunosc intrările *a, b* şi relaţia logică *f* şi se doreşte determinarea ieşirii *y*.

$$a, b, f \Rightarrow y \qquad (4.1)$$

Pentru cunoaşterea relaţiei logice *f* se aplică metoda *analitică*[43] care constă în descompunerea SLC în subsisteme cât mai simple şi stabilirea legăturilor dintre aceste subsisteme. În cazul SLC implementate cu contacte şi relee aceste subsisteme sunt de tipul *ŞI, SAU, NU*. Dacă SLC este realizat cu ajutorul circuitelor electronice integrate în afară de subsistemele menţionate mai sunt şi altele, de exemplu circuitele *ŞI-NU, SAU-NU, SAU-EXCLUSIV, etc.*

Dacă se cunosc elementele componente ale SLC şi structura[44] sa, rezultă imediat relaţia logică *f*.

Există mai multe metode pentru rezolvarea problemei analizei. În primul rând se poate calcula ieşirea direct folosind metodele algebrei booleene. Altă metodă constă în determinarea ieşirilor componentelor SLC, combinarea lor conform structurii şi urmărirea în continuare a modului cum se propagă semnalele de la intrarea SLC la ieşirea sa.

- ✓ ***Sinteza.***

Se cunosc intrările *a, b* şi ieşirea *y* şi se determină relaţia logică *f* care la rândul ei permite determinarea sistemului logic combinaţional. Sinteza este una dintre metodele inginereşti de proiectare pe lângă metoda exemplelor tip, metoda încearcă şi verifică şi altele.

$$a, b, y \Rightarrow f \qquad (4.2)$$

- ✓ ***Testarea.***

Se cunosc relaţia logică *f* şi ieşirea dorită *y* şi se determină intrările necesare *a* şi *b*.

$$f, y \Rightarrow a, b \qquad (4.3)$$

Această problemă trebuie rezolvată atunci când dorim să ştim dacă sistemul este defect, unde se găseşte defecţiunea şi care este tipul ei.

4.1. Coduri

Unele echipamente şi instalaţii au o funcţionare care poate fi descrisă printr-un număr finit de stări. De exemplu, un motor reversibil poate fi oprit, se poate roti în sens direct sau în sens direct, un ascensor se poate afla la unul dintre cele 12 etaje ale clădirii. Pentru acestea se foloseşte o codificare binară cu ajutorul mai multor biţi. Dacă avem *n* biţi putem codifica *N* stări cu ajutorul codului binar natural

$$N = 2^n \qquad (4.4)$$

[43] Aplicarea acestei metode se numeşte analiză

[44] Relaţiile dintre elemente.

De exemplu cele trei stări ale motorului reversibil pot fi codificate folosind un cod binar natura format din doi biţi *k1* şi *k2* care poate caracteriza, conform formulei (4.4), maximum patru stări. O codificare posibilă, nu singura, este prezentată în *Tab. 4.2*. Codul binar natural cu patru biţi este dat în *Tab. 4.4* iar codul binar distributiv cu patru biţi în *Tab. 4.3*. Se observă că codul distributiv este mai intuitiv dar are posibilităţi de codificare mult mai mici.

*Tab. 4.2 Codificarea cu doi biţi **a** şi **b** a stărilor unui motor reversibil*

Starea motorului	***a***	***b***
oprit	0	0
rotire directă	0	1
rotire inversă	1	0
-	1	1

Tab. 4.3 Codul binar distributiv sau "1 din n"

a	***b***	***c***	***d***
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0

Tab. 4.4 Codul binar natural cu patru biţi

a	***b***	***c***	***d***
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

4.2. Analiza sistemelor logice combinaţionale

Sistemele logice combinaţionale ***SLC*** sunt formate dintr-o mulţime de elemente, relaţii şi scopuri. Sistemele se numesc combinaţionale deoarece în funcţionarea lor nu intervine variabila timp. Contează numai combinaţia (structura) elementelor. Fără considerarea timpului aceste sisteme nu pot avea memorie şi deci nu pot realiza operaţii automate, fără intervenţia omului Elementele SLC pot fi caracterizate prin variabile binare care pot avea două valori {0,1}. Relaţiile dintre elemente sunt de tip *ŞI, SAU, NU* iar scopul acestor sisteme îl constituie prelucrarea informaţiei. Sistemele logice combinaţionale au drept model teoretic o algebră booleană şi în continuare ne vom folosi de rezultatele obţinute de acest model şi de terminologia specifică, Tab. 4.1. Elementele *SLC* sunt de natură electromecanică, electronică, hidraulică, etc. În continuare vom studia *SLC* formate din *contacte* şi *relee.* Variabilele binare ataşate contactelor vor

fi notate cu litere mici iar cele care caracterizează funcţionarea bobinelor, lămpilor, motoarelor, etc. sunt notate cu litere mari.

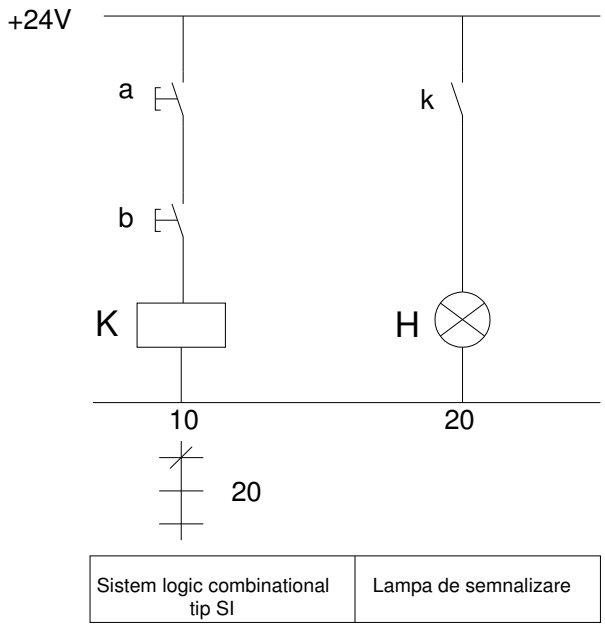


Fig. 4.2 Schema desfăşurată electrică a unui SLC de semnalizare tip ŞI

4.2.1. Sistem logic combinaţional tip ŞI

În Fig. 4.2 se prezintă un sistem logic combinaţional care are drept scop prelucrarea informaţiei pentru semnalizare. Lampa *H* se aprinde numai dacă butoanele *a ŞI* b sunt acţionate. Butonul este un contact normal deschis acţionat manual. Dacă este neacţionat prin el nu trece curent electric şi variabila care îl caracterizează are valoarea *0*. Atunci când este acţionat prin el trece curent şi variabila are valoarea *1*. Bobina releului *K* are şi ea două valori: *1* atunci când prin ea trece curent şi *0* atunci când prin ea nu trece curent electric. Valoarea ei este o funcţie de tip *ŞI* de valorile contactelor şi poate fi determinată *analizând* schema din Fig. 4.2. Prin *analiză* înţelegem că se cunoaşte SLC şi valorile semnalelor de intrare *a* şi *b* şi se determină teoretic, experimental sau prin simulare valorile semnalului de ieşire *K*. Pentru SLC cu contacte şi relee este foarte uşor să simulăm în imaginaţie funcţionarea sistemului pentru toate valorile posibile ale semnalelor de intrare şi să trecem valorile obţinute pentru semnalul de ieşire într-un tabel de adevăr cum este Tab. 4.5. Valorile contactelor *a* şi *b* pot fi trecute în orice ordine, important este să fie toate valorile posibile. Pentru a nu omite vre-o valoare vom folosi codul binar natural cu doi biţi. Acesta are 2^n valori, în care n este numărul de biţi. În cazul de faţă avem doi biţi *a* şi *b* şi numărul de valori ale codului este patru. În primul rând al Tab. 4.5 contactele au valoarea *0* adică sunt neacţionate. Prin ele nu poate să treacă curent electric şi deci şi curentul care trece prin bobina releului este nul. Deci rezultatul simulării indică valoarea *0* pentru variabila *K* care este

trecută în ultima coloană. Se procedează asemănător pentru toate situaţiile în care unul sau altul dintre butoane sunt acţionate.

Tab. 4.5 Tabel de adevăr cu rezultatele analizei SLC tip ŞI

a	***b***	***K***
0	0	**?….0**
0	1	**?….0**
1	0	**?….0**
1	1	**?….1**

Releul electromagnetic are, într-o primă aproximare în care nu intervine timpul, următoarea relaţie dintre mărimea sa de intrare, starea *K* a bobinei şi mărimea sa de ieşire, contactul normal deschis *k*:

$$k = K \tag{4.5}$$

Această relaţie spune că dacă trece curentul prin bobină şi deci releul este acţionat atunci contactul său normal deschis este tot acţionat. Considerând Tab. 4.5 şi (4.5) se poare definii relaţia logică de tip *ŞI* între contactele de intrare *a* şi *b* ale SLC şi contactul său de ieşire *k*.

$$k = a \cdot b \tag{4.6}$$

Conform Tab. 4.1 această operaţie mai este numită şi înmulţire booleană iar operatorul punct nu este specificat adeseori.

Funcţia logică *ŞI* corespunde structurii de conectare în serie a contactelor.

Între lampa de semnalizare din Fig. 4.2 caracterizată prin variabila *H* şi contactul releului *k* există relaţia *H=k* adică lampa *H* este aprinsă atunci când contactul *k* este acţionat şi considerând (4.6) se stabileşte relaţia logică a SLC de semnalizare prin acţionarea manuală a butoanelor: *H=a.b.*

4.2.2. Sistem logic combinaţional tip SAU

Contactele legate în paralel formează un sistem logic combinaţional de tip SAU. Un exemplu cu două contacte este prezentat în *Fig. 4.3*. Analizând acest sistem prin simulare în imaginaţie într-un mod asemănător ca şi în cazul SLC tip ŞI se obţine tabelul de adevăr Tab. 4.6. Lampa *H* este acţionată (aprinsă) numai dacă sunt acţionate butoanele *a SAU b.*

Tab. 4.6 Tabelul de adevăr cu rezultatele SLC tip SAU

a	***b***	***K***
0	0	**?….0**
0	1	**?….1**
1	0	**?….1**
1	1	**?….1**

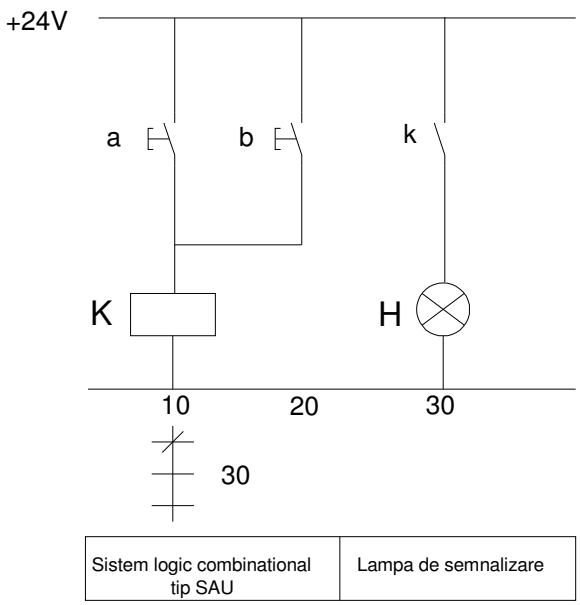


Fig. 4.3 Schema desfăşurată electrică pentru un SLC de semnalizare tip SAU

Relaţia logică între contactele de intrare *a* şi *b* ale sistemului şi ieşirea sa *k* tip contact este:

$$k = a + b \tag{4.7}$$

4.2.3. Sistem logic combinaţional tip NU

Sistemul logic combinaţional tip NU are un singur contact normal închis ca în *Fig. 4.4*. Rezultatul analizei prin simulare este prezentat în Tab. 4.7.

Tab. 4.7 Tabelul de adevăr cu rezultatele analizei SLC tip NU

a	***K***
0	**?….1**
1	**?….0**

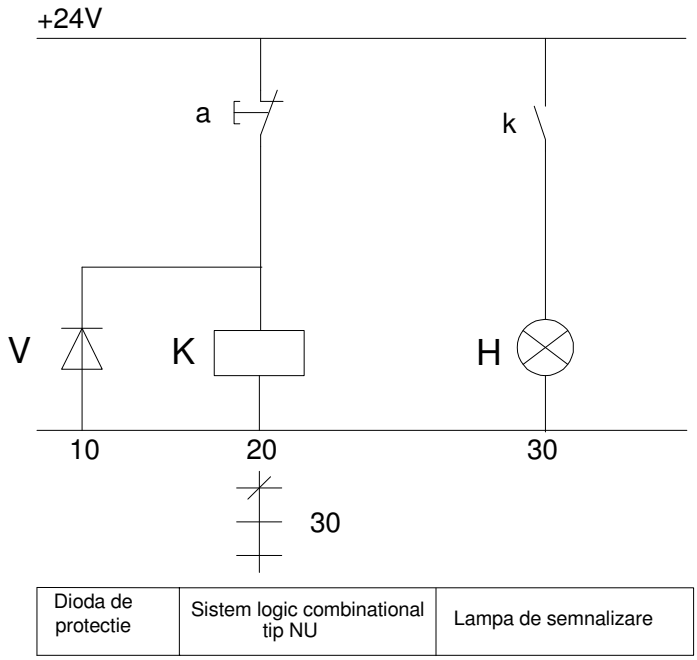


Fig. 4.4 Schema desfăşurată electrică pentru SLC de semnalizare tip NU

Dioda din circuitul 10 nu face parte dintre elementele SLC. Reamintim că acestea sunt caracterizate prin două stări. Aceasta este doar o aproximare. În realitate trecerea de la o stare la alta nu se face brusc ci în cadrul unui proces tranzitoriu care poate avea efecte nedorite. Acesta este cazul sarcinii inductive formate din bovina releului din circuitul 20. La deschiderea contactului *a* are loc un fenomen de autoinducţie care produce o tensiune foarte mare care se opune tensiunii de alimentare. această tensiune provoacă scântei la contactul *a* care produc paraziţi şi deteriorează contactul.

Relaţia logică intrare – ieşire cu semnale de tip contact este:

$$k = \bar{a} \tag{4.8}$$

4.2.4. Relaţii logice caracteristice sistemelor logice combinaţionale

Deoarece modelul teoretic al unui sistem logic combinaţional este o algebră booleană între elementele sale componente există următoarele relaţii logice care pot fi verificate prin simulare.

- ✓ Comutativitate

$$ab = ba \qquad a + b = b + a \tag{4.9}$$

- ✓ Asociativitate

$$a(bc) = (ab)c \qquad a + (b + c) = (a + b) + c \tag{4.10}$$

- ✓ Distributivitate

$$a(b+c)=ab+ac \qquad a+bc=(a+b)(a+c) \tag{4.11}$$

- ✓ Absorbție

$$a+ab=a \qquad a(a+b)=a \tag{4.12}$$

- ✓ Idempotență

$$aa=a \qquad a+a=a \tag{4.13}$$

- ✓ Dublă negație

$$\overline{\overline{a}}=a \tag{4.14}$$

- ✓ Teorema lui De Morgan

$$\overline{ab}=\overline{a}+\overline{b} \qquad \overline{a+b}=\overline{a}\,\overline{b} \tag{4.15}$$

- ✓ Terțul exclus

$$a+\overline{a}=1 \qquad a\overline{a}=0 \tag{4.16}$$

- ✓ Relații cu constante logice

$$a0=0 \qquad a+0=a \tag{4.17}$$

$$a1=a \qquad a+1=1 \tag{4.18}$$

$$\overline{1}=0 \qquad \overline{0}=1 \tag{4.19}$$

4.2.5. Elemente reale ale sistemelor logice combinaționale

Elementele SLC sunt aproximate prin modele teoretice ideale care au numai două stări. În realitate lucrurile nu stau aşa. De exemplu contactele sunt considerate drept elemente care lasă să treacă sau întrerup total curentul electric. Contactul real are însă o rezistență electrică cu o valoare cuprinsă între 0,01 şi 10 Ω. Valorile mari ale rezistenței de contact sunt atinse după o funcționare mai îndelungată şi conduc, în cazul unor curenți mari, la căderi de tensiune importante care periclitează acționarea releului. Acest fenomen limitează numărul contactelor conectate în serie, fenomen care nu este luat în considerare de modelul teoretic sub formă de algebră booleană al sistemelor logice combinaționale.

Caracteristica statică de acționare a releului nu este nici ea simplă. Forma ei este de tip histerezis şi trebuie luată în considerare la proiectarea sistemelor logice combinaționale.

Un alt fenomen care apare în cazul sistemelor logice combinaționale este hazardul combinațional prezentat în Fig. 4.5. Conform relației logice (4.16) lampa H ar trebui să fie tot timpul aprinsă deoarece atunci când un contact *a* se deschide alt contact *a* se închide instantaneu. În realitate acțiunile instantanee nu sunt posibile în practică. Este posibil ca închiderea contactului a să se facă mai încet decât deschiderea celuilalt contact a. În aceste condiții lampa H se aprinde pentru o fracțiune de secundă. Această pâlpâire poate trece neobservată, dar în alte situații

impulsul scurt generat de hazardul combinaţional să poată fi memorat de alte sisteme şi transformat într-un impuls lung cu efecte dăunătoare.

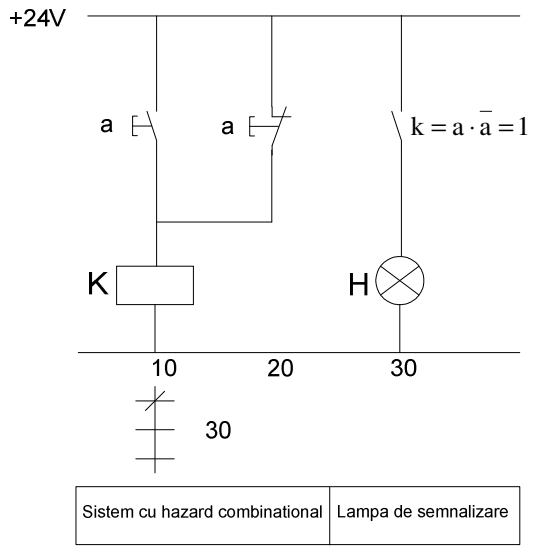


Fig. 4.5 Hazardul combinaţional la un element tip SAU.

4.3. Sinteza sistemelor logice combinaţionale

Sinteza determinată de (4.2) este una dintre metodele de proiectare ale sistemelor logice combinaţionale.

4.3.1. Metoda formei disjunctive canonice

Să considerăm sistemul logic combinaţional pentru aprinderea unei lămpi de la capetele unui coridor, Fig. 4.6.

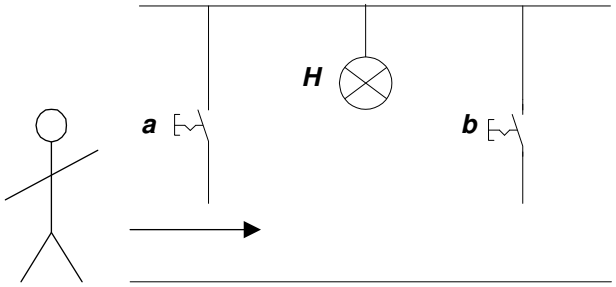


Fig. 4.6 Schema tehnologică pentru aprinderea unei lămpi de la capetele unui coridor

Semnalele de intrare în SLC sunt variabilele binare ataşate butoanelor *a* şi *b* cu memorie mecanică iar semnalul de ieşire este variabila binară ataşată lămpii *H*. Modul dorit de funcţionare a SLC este prezentat în Tab. 4.8. Dacă omul nu se găseşte în coridor şi butoanele *a, b* nu sunt apăsate, Fig. 4.6, atunci lampa *H* este stinsă, situaţie specificată în prima linie a tabelului de adevăr Tab. 4.8.

Tab. 4.8 Tabelul de adevăr pentru SLC tip SAU-EXCLUSIV

a	***b***	***H=f(a,b)***
0	0	**f(0,0)=0**
0	1	**f(0,1)=1**
1	0	**f(1,0)=1**
1	1	**f(1,1)=0**

La intrarea în coridor omul acţionează butonul *a* cu memorie mecanică. Lampa *H* trebuie să se aprindă, caz prezentat în linia trei a Tab. 4.8. Omul avansează pe coridor dar butonul *a* rămâne acţionat deoarece a memorat impulsul de acţionare. La capătul coridorului omul acţionează prin apăsare butonul *b* care rămâne în această stare deoarece are memorie. Lampa se stinge. Situaţia este specificată în linia patru a tabelului de adevăr Tab. 4.8. La întoarcere situaţia se repetă. La început ambele butoane sunt apăsate şi lampa stinsă (linia patra). La intrarea în coridor se acţionează butonul *b*. Acesta fiind în starea de memorare a acţionării precedente este apăsat. Prin noua acţionare nu mai este apăsat iar lampa trebuie să se aprindă, situaţie reflectată în linia a doua a Tab. 4.8. În continuare valorile se repetă.

Cunoaştem acum intrările şi ieşirea dar nu ştim funcţia logică *f* care le leagă. Pentru a o afla presupunem că relaţia logică este de forma:

$$H = f(a,b) = ua + v\bar{a} \qquad (4.20)$$

în care *u* şi *v* sunt două funcţii logice numai de variabila *b*.

Dar variabila *a* poate lua numai două valori, 0 sau 1. Să examinăm cele două cazuri pe rând.

$$a = 0 \qquad H = f(0,b) = u \cdot 0 + v \cdot 1 = v \qquad (4.21)$$

$$a = 1 \qquad H = f(1,b) = u \cdot 1 + v \cdot 0 = u \qquad (4.22)$$

Deci (4.20) devine:

$$H = f(a,b) = f(1,b)a + f(0,b)\bar{a} \qquad (4.23)$$

Repetând procedeul pentru *f(1,b)* şi *f(0,b)* se obţine *forma disjunctivă canonică* a relaţiei logice:

$$H = f(a,b) = f(0,0)\bar{a}\bar{b} + f(0,1)\bar{a}b + f(1,0)a\bar{b} + f(1,1)ab \qquad (4.24)$$

Se observă că coeficienţii variabilelor *a* şi *b* sunt tocmai valorile lui *H* din Tab. 4.8. Înlocuind se obţine:

$$H = \bar{a}b + a\bar{b} \tag{4.25}$$

Această relaţie poate fi implementată foarte uşor dacă se ţine seama că funcţia *SAU* înseamnă conectarea butoanelor în paralel, funcţia *ŞI* înseamnă conectarea butoanelor în serie, iar funcţia *NU* se realizează cu un buton având contactul normal închis. Proiectarea SLC trebuie să ţină seama însă şi de alte considerente, nu numai relaţia logică între intrări şi ieşiri. Dacă, de exemplu, se doreşte ca curentul prin butoane să fie mult mai mic decât curentul de sarcină al lămpii atunci se foloseşte un releu intermediar şi de obţine schema din *Fig. 4.7*. Analiza acestui SLC arată că este format din două subsisteme. Primul este format din bobina releului şi butoanele din circuitele 10 şi 20 iar relaţia logică dintre intrări şi ieşiri (4.26) este de tipul (4.25) realizând funcţia tip *SAU-EXCLUSIV*. Al doilea subsistem este format din lampa şi contactul *k* al releului care se găsesc în circuitul 30. Relaţia logică a acestui subsistem este (4.27) care adăugată la relaţia fizică (4.28) permite realizarea relaţiei dorite (4.25) după eliminarea variabilelor intermediare *k* şi *K*.

$$K = \bar{a}b + a\bar{b} \tag{4.26}$$

$$H = k \tag{4.27}$$

$$k = K \tag{4.28}$$

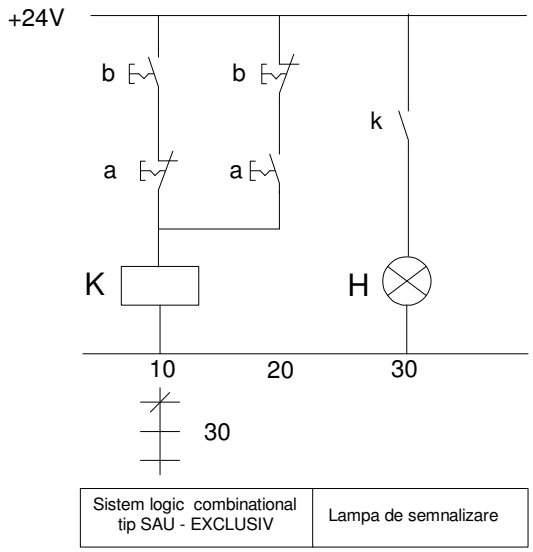


Fig. 4.7 Schema desfăşurată electrică a sistemului logic combinaţional pentru aprinderea unei lămpi de la capetele unui coridor

4.3.2. Metoda diagramei Karnaugh

Sinteza sistemelor logice combinaţionale prin metoda formei disjunctive canonice conduce, în general, la relaţii logice complexe atunci când numărul

variabilelor de intrare este mare. Acest fapt poate fi ilustrat de exemplul SLC de semnalizare a majorităţii cu caietul de sarcini în Tab. 4.9. Lampa de semnalizare H se aprinde atunci când majoritatea butoanelor sunt acţionate. Atunci când avem trei butoane de intrare majoritatea este formată de două dintre ele.

Tab. 4.9 Tabelul de adevăr al SLC de semnalizare a majorităţii

a	***b***	***c***	***H=f(a,b,c)***
0	0	0	**f(0,0,0)=0**
0	0	1	**f(0,0,1)=0**
0	1	0	**f(0,1,0)=0**
0	1	1	**f(0,1,1)=1**
1	0	0	**f(1,0,0)=0**
1	0	1	**f(1,0,1)=1**
1	1	0	**f(1,1,0)=1**
1	1	1	**f(1,1,1)=1**

Forma disjunctivă canonică a relaţiei logice dintre intrările şi ieşirea SLC caracterizat de Tab. 4.9 este de tipul (4.24) cu deosebirea că de data aceasta există trei semnale de intrare.

$$H = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc \qquad (4.29)$$

Implementarea aceste relaţii cu contacte şi relee necesită 12 butoane dintre care trei sunt prevăzute cu contacte normal închise.

H

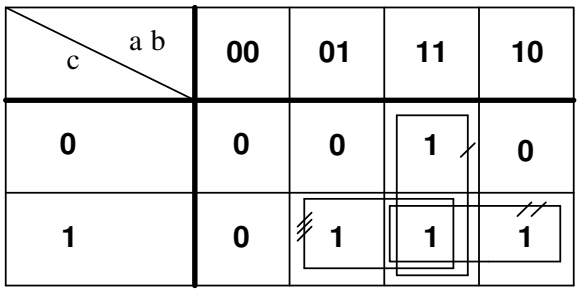


Fig. 4.8 Diagrama Karnaugh pentru SLC de semnalizare a majorităţii

In relaţia (4.29) se poate da factor comun variabila a şi se obţine o formă simplificată:

$$H = \bar{a}bc + a[\bar{b}c + b\bar{c} + bc] \qquad (4.30)$$

Implementarea acestei relaţii necesită numai 10 contacte. Alte operaţii caracteristice algebrei booleene ar putea produce simplificări mai importante. Există însă o alternativă care conduce la simplificarea maximă posibilă. Aceasta este metoda diagramei Karnaugh prezentată în *Fig. 4.88*. Diagrama Karnaugh nu

este nimic altceva decât tabelul de adevăr altfel scris. Capul de tabel al diagramei Karnaugh este scris în cod Gray la care trecerea de la o valoare la alta se face prin modificarea unui singur bit. Odată ce tabelul de adevăr a fost rescris sub forma diagramei Karnaugh se realizează următoarele etape pentru obținerea relației logice.

- ✓ ***Gruparea celulelor.*** Toate celulele diagramei Karnaugh care conțin valoarea logică 1 se grupează după următoarele reguli.

R1- O grupare poate conține un număr N de celule care conțin valoarea 1 care este egal cu $N=2^n$, în care n = 0,1,2,3,4,5…Practic o grupare va conține 1, 2, 4, 8, 16 celule. Se observă că o grupare poate avea o singură celulă. Grupările se numerotează așa cum s-a procedat în *Fig. 4.88*.

R2 – Grupările trebuie să fie cât mai mari şi cât mai puține.

R3 – Toate celulele dintr-o grupare trebuie să aibă câte o latură comună, nu un nod comun. Din această cauză grupările au forma unor dreptunghiuri sau pătrate şi nu a unei stele

R4 – O celulă poate să facă parte din mai multe grupări.

R5 – Suprafața în care se găsesc celulele diagramei Karnaugh ar laturile opuse lipite. În felul acesta se pot face grupări cu celulele de graniță sau celulele situate la colțuri.

- ✓ ***Relația logică cu variabile ondulate.*** Pentru fiecare grupare se scrie produsul variabilelor ondulate. Aceste produse se adună pentru a obține variabila de ieşire. Pentru *Fig. 4.88* se obține:

$$H = \underline{\tilde{a}\,\tilde{b}\,\tilde{c}} + \underline{\underline{\tilde{a}\,\tilde{b}\,\tilde{c}}} + \underline{\underline{\underline{\tilde{a}\,\tilde{b}\,\tilde{c}}}} \qquad (4.31)$$

- ✓ ***Calculul variabilelor ondulate.*** Să considerăm o variabilă ondulată oarecare, să zicem ***ã***. Pentru gruparea considerată se observă ce valori are variabila neondulată ***a*** pentru fiecare dintre celulele grupării. Există trei cazuri. În primul rând este posibil ca variabila neondulată să aibă pentru toate celulele grupării valoarea unu. Atunci variabila ondulată este egală cu variabila neondulată.

$$a = 1 \quad \Rightarrow \quad \tilde{a} = a \qquad (4.32)$$

Asemănător

$$a = 0 \quad \Rightarrow \quad \tilde{a} = \overline{a} \qquad (4.33)$$

Dacă pentru o grupare variabila neondulată are pentru unele celule valoarea unu ***şi*** pentru alte celule valoarea zero atunci variabila ondulată este o constantă logică egală cu unu:

$$[a = 1] \bigcap [a = 0] \Rightarrow \tilde{a} = 1 \qquad (4.34)$$

Aplicând aceste reguli la diagrama Karnaugh din *Fig. 4.88* se obține relația logică dorită:

$$H = ab + ac + bc \tag{4.35}$$

Dacă se dă factor comun o variabilă, de exemplu a, se obține o nouă simplificare:

$$H = a(b + c) + bc \tag{4.36}$$

Implementarea relației logice (4.36) folosind un releu intermediar pentru a obține curenți mici prin contacte este prezentată în Fig. 4.99. În acest exemplu folosirea metodei de sinteză a diagramei Karnaugh a permis simplificarea schemei, comparativ cu cea obținuta la metoda formei disjunctive canonice, de la 12 la 5 contacte.

Exemplul precedent a arătat că sinteza SLC necesită cunoaşterea tuturor intrărilor şi ieşirilor corespunzătoare. În practică multe valori ale intrărilor nu pot apare din motive tehnologice sau ieşirile pentru ele sunt indiferente. Chiar dacă am ştii toate valorile intrărilor nu este posibilă prelucrarea lor practică deoarece numărul lor creşte exponențial cu numărul semnalelor de intrare. Pentru n semnale de intrare avem $N=2^n$ valori ale semnalelor de ieşire. O manieră de a depăşi aceste dificultăți este prezentată în exemplul următor.

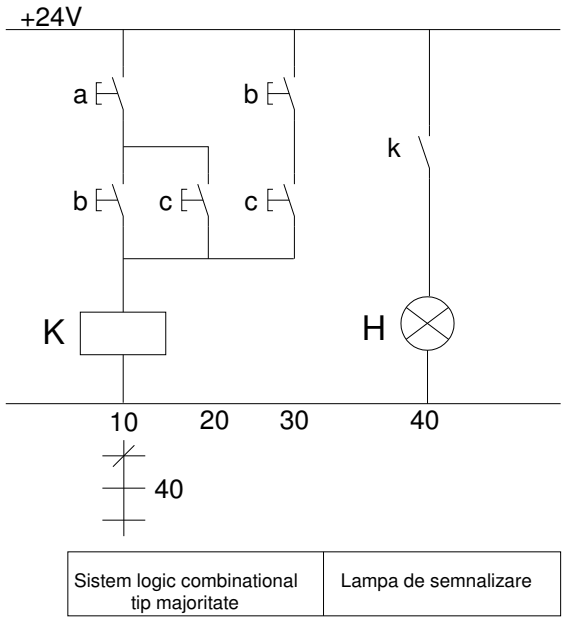


Fig. 4.9 Schema desfăşurată electrică pentru SLC cu majoritate

t

d e \ b c	**00**	**01**	**11**	**10**
00	0	0	—	0
01	0	0	—	1
11	0	1	—	1
10	0	0	—	1

Fig. 4.10 Diagrama Karnaugh a SLC pentru semnalizarea suprasarcinii

Se doreşte sinteza unui sistem logic combinaţional care să semnalizeze apariţia suprasarcinii unui generator de 100 kW. Sarcina generatorului este formată din cinci motoare: motorul *a* de 51 kW, motorul *b* de 40 de kW, motorul *c* de 20 kW, motorul *d* de 20 kW şi motorul *e* de 10 kW. Motoarele *b* şi *c* nu funcţionează niciodată simultan din motive tehnologice. Sistemul având cinci semnale logice de intrare *a,b,c,d* şi *e,* numărul valorilor posibile este de 2^5=32. Putem evita considerarea tuturor acestor valori dacă descompunem sistemul format din cele cinci motoare în două subsisteme: motorul *a* şi grupul motoarelor *b,c,d,e*. Se observă că suprasarcina, adică consumul mai mare de 100kW, apare numai dacă motorul *a* funcţionează *ŞI* al doilea subsistem consumă mai mult de 50 kW.

Deci lampa de semnalizare H se aprinde dacă:

$$H = a \cdot t \tag{4.37}$$

Să notăm cu *t* condiţia ca al doilea subsistem să consume mai mult de 50 kW. Diagrama Karnaugh pentru variabila de ieşire *t* a subsistemului format din cele patru motoare este prezentată în Fig. 4.1010. Relaţia logică corespunzătoare este:

$$t = \tilde{b}\tilde{c}d\tilde{e} + b\tilde{c}d\tilde{e} + b\tilde{c}\tilde{d}\tilde{e} = b11e + b1d1 + 1cde = b(e+d) + cde \tag{4.38}$$

Considerând (4.37) rezultă relaţia logică pentru ieşirea SLC formată din lampa H.

$$H = at = a[b(e+d) + cde] \tag{4.39}$$

Dacă se doreşte semnalizarea funcţionării normale a generatorului electric, fără suprasarcină, atunci lampa H1 se va aprinde atunci când *NU* se aprinde H. Din (4.37) se obţine:

$$H1 = \overline{H} = \overline{a \cdot t} = \bar{a} + \bar{t} \qquad (4.40)$$

Diagrama Karnaugh pentru complementul lui *t* se obţine din Fig. 4.1010 în care se iau toate valorile negate ca în *Fig. 4.111*.

După ce se fac grupările şi se calculează variabilele ondulate se obţine relaţia:

$$\bar{t} = \overline{bc} + \overline{de} + \overline{bd} + \overline{be} \qquad (4.41)$$

Din (4.40) rezultă condiţia de aprindere a lămpii H1 la funcţionarea normală a generatorului:

$$H1 = \bar{a} + \bar{t} = \bar{a} + \bar{b}(\bar{c} + \bar{e}) + \bar{d}(\bar{e} + \bar{b}) \qquad (4.42)$$

Relaţia (4.41) se poate obţine şi din (4.38) dacă se aplică teorema lui De Morgan şi se consideră relaţia (4.18).

$$\bar{t} = \overline{be + bd + cde} = (\overline{be})(\overline{bd})(\overline{cde}) = (\bar{b} + \bar{e})(\bar{b} + \bar{d})(\bar{c} + \bar{d} + \bar{e}) \qquad (4.43)$$

$$\bar{t} = (\bar{b} + \overline{bd} + \overline{eb} + \overline{ed})(\bar{c} + \bar{d} + \bar{e}) = \overline{bc} + \overline{de} + \overline{bd} + \overline{be} \qquad (4.44)$$

$\bar{t}$

de \ bc	00	01	11	10
00	1	1	—	1
01	1	1	—	0
11	1	0	—	0
10	1	1	—	0

Fig. 4.11 Diagrama Karnaugh a SLC de semnalizare a funcţionării fără suprasarcină a generatorului

4.4. Testarea sistemelor logice combinaţionale

Problema testării a fost definită de (4.3). Dacă dorim să testăm SLC din Fig. 4.122 trebuie să cunoaştem o valoare a ieşirii. În cazul de faţă dorim ca lampa să fie aprinsă, deci Z=1. Pentru a testa dacă SLC aprinde corect lampa este nevoie să ştim care sunt semnalele de intrare pentru care lampa este aprinsă. Aceasta se rezolvă uşor dacă cunoaştem relaţia logică dintre intrări şi ieşire. Deoarece de obicei se cunoaşte schema desfăşurată electrică, ca în Fig. 4.122, se face mai întâi o analiză SLC din care rezultă (4.45).

$$Z = g + h = \bar{a}c + bc = c(\bar{a} + b) \tag{4.45}$$

Cu ajutorul relaţiei logice intrări – ieşire se determină tabelul de adevăr prezentat în *Tab. 4.10*. Din acesta se constată care sunt intrările pentru care ieşirea are valoarea unu, adică lampa este aprinsă. Acestea sunt

$$(a,b,c) = (0,0,1) \tag{4.46}$$

$$(a,b,c) = (0,1,1) \tag{4.47}$$

$$(a,b,c) = (1,1,1) \tag{4.48}$$

Relaţiile (4.46), (4.47) şi (4.48) spun că sistemul logic combinaţional din Fig. 4.122 poate fi testat dacă este defect prin apăsarea butonului *a* sau apăsarea simultană a lui *a* şi *b* sau apăsarea simultană a tuturor celor trei butoane *a,b,c*. Această metodă de testare pune în evidenţă faptul că a apărut o defecţiune atunci când lampa nu se aprinde, dar nu oferă nici o indicaţie asupra locului unde a apărut defecţiunea.

Examinând manevra (4.46) se constată că la apăsarea butonului *c* lampa Z se aprinde prin intermediul releului G. Dacă lampa nu se aprinde înseamnă că releul este stricat şi am localizat defecţiunea în circuitul cu numărul 10. În mod asemănător rezultă că dacă lampa Z nu se aprinde la apăsarea tuturor celor trei butoane releul H din circuitul 20 este defect. Semnalul de intrare (4.29) nu permite localizarea defecţiunii deoarece lampa se poate aprinde sau prin releul G sau prin releul H. Dacă unul dintre relee este defectat lampa tot se prind prin intermediul celuilalt releu şi defecţiunea nu este pusă în evidenţă.

Tab. 4.10 Tabelul de adevăr pentru SLC din Fig. 4.12 care este testat

a	***b***	***c***	***Z***
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**0**
1	1	0	**0**
1	1	1	**1**

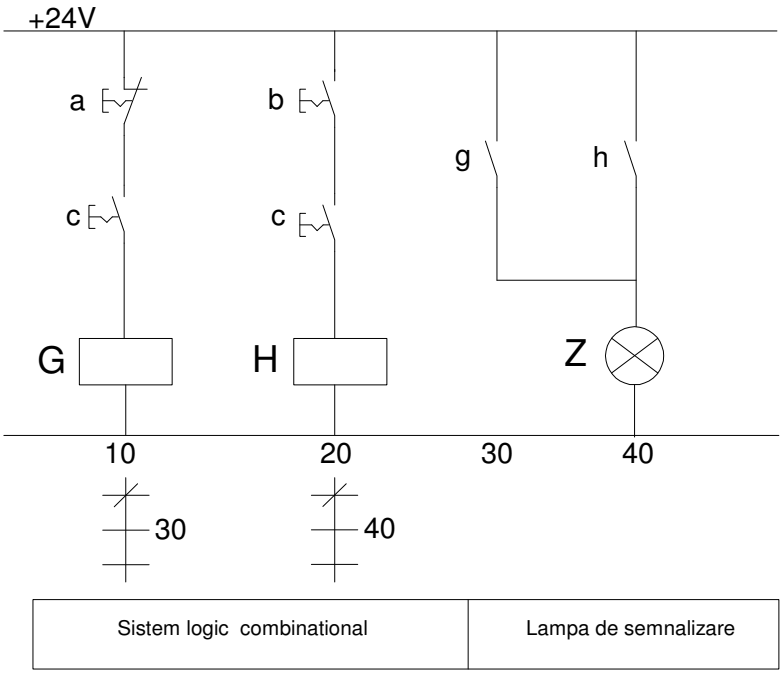


Fig. 4.12 Sistem logic combinaţional supus testării

La sistemele logice combinaţionale realizate cu contacte şi relee tipurile cele mai frecvente de defecţiuni îl constituie scurtcircuitele la masă sau la borna de alimentare care conduc la înţepenirea unor relee pe valoarea acţionat sau valoarea neacţionat. Depistarea tipului de defect se poate face cu ajutorul relaţiilor logice diferenţiale.

4.5. Sisteme logice combinaţionale cu circuite integrate

Sistemele logice combinaţionale pot fi implementate software, cu ajutorul calculatoarelor sau automatelor programabile sau hardware folosind circuite electrice, electronice, fluidice şi altele. Implementarea cu circuite electronice integrate foloseşte pentru elementele de bază simbolurile din Fig. 4.133. Pornind de la aceste simboluri se formează şi altele. De exemplu, simbolurile pentru circuitele logice combinaţionale *ŞI-NU* şi *SAU-NU* au câte un cerculeţ la ieşire după modelul circuitului *NU*.

În cazul circuitelor electronice integrate semnalele de intrare şi ieşire sunt tensiuni care pot lua numai două valori: zero volţi pentru zero logic şi cinci volţi pentru unu logic.

Simbolurile din Fig. 4.133 sunt importante şi pentru programarea automatelor programabile.

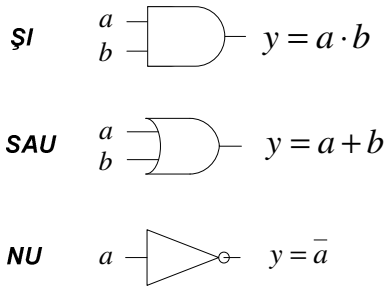


Fig. 4.13 Simbolurile pentru circuitele ŞI, SAU, NU.

Un exemplu de folosire a simbolurilor pentru circuite integrate este prezentat în Fig. 4.144.

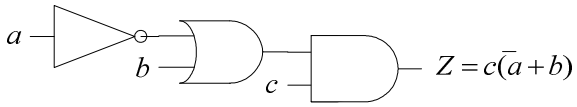


Fig. 4.14 Implementarea cu circuite integrate a relaţiei (4.45)

5. Sisteme cu evenimente discrete

Evenimentele discrete apar în mod frecvent la organizarea serviciilor oferite de clădiri şi locuinţe, în funcţionarea instalaţiilor, echipamentelor informatice şi a atelierelor de producţie. Din această cauză sistemele cu evenimente discrete sunt foarte utile în luarea deciziilor referitoare la diferite aspecte ale activităţii inginereşti, economice şi organizatorice legate de clădiri şi locuinţe. *Sistemele discrete logice* sunt cazuri particulare, mai simple, de sisteme cu evenimente discrete şi sunt folosite intensiv la conducerea automată a instalaţiilor şi echipamentelor din clădiri şi locuinţe. Ele vor fi considerate drept o *generalizare* a sistemelor logice combinaţionale[45] *SLC* sau o *particularizare* a sistemelor cu evenimente discrete[46] SED.

Trei *modele* importante folosite în analiza şi sinteza sistemelor discrete logice sunt reţeaua Petri interpretată sigură, maşina de stări şi grafcetul.

Spre deosebire de sistemele logice combinaţionale sistemele discrete logice au o comportare *dinamică,* adică semnalele lor de intrare şi ieşire depind de timp, şi au *memorie* pentru că semnalele lor de ieşire depind de starea precedentă. Legătura dintre semnalele de intrare şi semnalele de ieşire *logice* ale unui astfel de sistem este prezentată adeseori *grafic* sub forma unei reţele Petri[47] interpretate sigure, a unui graf de comandă etapă-tranziţie GrafCET[48] sau a unui grafic numit maşină de stări.

Sistemele discrete logice se deosebesc, în principal, de sistemele cu evenimente discrete prin faptul că mărimile de intrare, ieşire şi stare sunt logice, adică pot avea numai două valori: 0 sau 1.

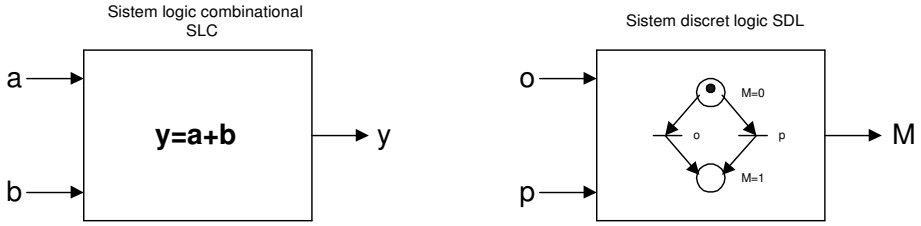


Fig. 5.1 Exemple de relaţii intrare-ieşire la un SLC şi la un SDL

La sistemele logice combinaţionale relaţia intrare-ieşire poate fi reprezentată sub forma unui tabel de adevăr sau a unei funcţii logice, ca în Fig. 5.1 a).

La sistemele discrete logice aceiaşi relaţie intrare – ieşire poate fi reprezentată ca în Fig. 5.1 b) de un graf, o reţea Petri interpretată sigură în acest exemplu, sau de un sistem de funcţii logice de timp.

---

[45] Sisteme logice combinaţionale generalizate prin introducerea unei reacţii (feedback).

[46] Sisteme cu evenimente discrete particularizate prin admiterea numai a valorilor logice pentru stări.

[47] Concept elaborat de Carl Adam Petri în 1962 în Germania.

[48] Concept elaborat de o comisie în 1979 în Franţa.

Sistemele cu evenimente discrete pot realiza câteva operaţii importante pentru conducerea sistemelor[49]: secvenţialitatea, paralelismul general, paralelismul conveier, concurenţa, interblocarea şi sincronismul.

Problemele sistemelor logice combinaţionale, analiza, sinteza (proiectarea), testarea, optimizarea şi implementarea se regăsesc şi la sistemele cu evenimente discrete.

Analiza sistemelor discrete logice se face pornind de la faptul că acestea sunt cazuri particulare ale sistemelor cu evenimente discrete. Se distinge analiza structurală de analiza comportamentală a acestor sisteme.

In esenţă, proiectarea sistemelor discrete logice folosite în conducerea instalaţiilor şi proceselor din clădiri se va baza pe proiectarea sistemelor logice combinaţionale.

Implementarea modelelor stabilite pentru sistemele discrete logice este mai dificilă decât în cazul sistemelor logice combinaţionale. Se va pune accentul pe implementarea cu contacte şi relee şi implementarea cu automate programabile logice.

5.1. Modelarea sistemelor cu evenimente discrete

Sistemele discrete logice pot fi considerate drept cazuri particulare de sisteme cu evenimente discrete şi din această cauză vom studia, la început, cazul, mai general, al modelării sistemelor cu evenimente discrete.

Există multe modele ale sistemelor cu evenimente discrete[50]. In domeniul sistemelor cu conducere automată se folosesc drept modele foarte mult *reţelele Petri interpretate sigure*[51] care sunt nişte cazuri mai simple, de reţele Petri. Vom folosi acest tip de modele atât pentru sistemul automat[52], în întregul său, cât şi pentru partea din sistemul automat care este formată din echipamentul de conducere.

In *Fig. 5.2* se prezintă reţeaua Petri interpretată care modelează sub forma unui sistem cu evenimente discrete SED funcţionarea unei instalaţii electrice formată din trei resurse: echipamentul de alimentare al motorului, echipamentul de alimentare al automatului motorului şi motorul electric. Intr-o primă aproximaţie[53] se disting trei stări in funcţionarea acestui SED: nealimentat, alimentat şi oprit, pornit. Stările sistemului sunt reprezentate în reţeaua Petri sub forma unor cercuri numite *poziţii* sau locaţii. Reţeaua Petri din *Fig. 5.2* are trei poziţii: P1, P2 şi P3. Resursele unei stări a SED sunt reprezentate sub forma unor *mărci* sau jetoane prezente în poziţie. De exemplu, poziţia P1 din exemplul considerat are două resurse: echipamentul de alimentare al automatului şi echipamentul de alimentare şi protecţie a motorului. O stare este activă dacă dispune de resurse, adică există mărci în interiorul cercului care o reprezintă. Dacă starea este activă se execută

[49] Instalaţiilor.

[50] Dintre aceste modele menţionăm: limbajele naturale, limbaje formale, automate stochastice temporizate, maşini de stări, reţele Petri şi grafuri de evenimente, lanţuri Markov, reţele de cozi de aşteptare, diagrame ale fluxurilor de date, etc.

[51] Se mai numesc reţele Petri interpretate logice sau reţele Petri interpretate binare..

[52] Instalaţia automatizată.

[53] Pentru motoarele de putere pornite după algoritmul stea-triunghi există mai multe stări.

acţiunile care o interpretează şi sunt prezentate în dreptul poziţiei sale. Pentru instalaţia din *Fig. 5.2* prezentată în momentul iniţial există o singură stare activă, poziţia P1 şi se execută următoarele acţiuni: automatul şi motorul nu sunt alimentate. Trecerea de la o stare activă la altă stare activă a SED se face prin intermediul unor *arce orientate ponderate* si a unor *tranziţii* reprezentate sub forma unor *dreptunghiuri înnegrite*[54]. In *Fig. 5.2* există trei tranziţii: T1, T2 şi T3. Tranziţiile se *declanşează*[55] la apariţia unor evenimente externe, dacă există resursele necesare, adică sunt *validate*. Evenimentele ataşate tranziţiilor T1, T2 şi T3 sunt acţionarea în impuls[56] a butoanelor: *b* de alimentare, *p* de pornire şi *o* de oprire. După declanşare se consumă resurse din poziţiile precedente şi se creează resurse în poziţiile posterioare. Numărul resurselor consumate şi create este dat de ponderea arcelor.

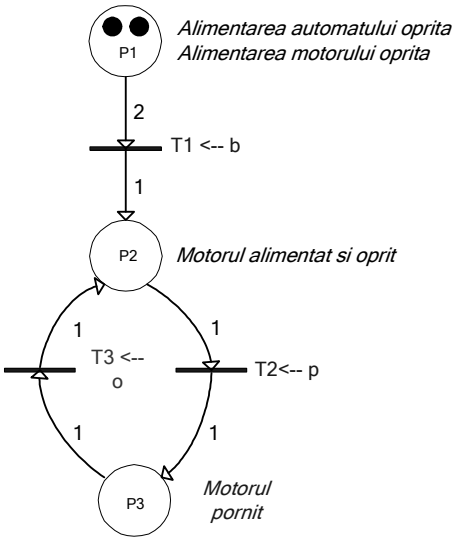


Fig. 5.2 O reţea Petri interpretată[57] care modelează sistemul de conducere al unui motor electric.

De exemplu, pentru situaţia din *Fig. 5.2* dacă apare evenimentul acţionării butonului de alimentare, adică *b = 1*, tranziţia T1 la care acest eveniment o interpretează se declanşează şi se consumă două mărci din poziţia P1 si se creează

[54] Un exemplu clasic de reprezentare a tranziţiilor este prezentat în *Fig. 5.6*. In alte figuri tranziţiile sunt reprezentate prin dreptunghiuri foarte subţiri, mai degrabă linii îngroşate, în maniera folosită de grafcet.

[55] Fire (eng).

[56] Butoanele nu au memorie mecanică.

[57] Reţeaua este *nesigură* pentru că în P1 sunt mai multe mărci.

o marcă în poziția P2. Se obține rețeaua Petri din Fig. 5.3. Dacă în aceiași situație se apasă butonul de pornire *p* sau butonul de oprire *o* tranzițiile la care sunt atașate aceste evenimente nu se declanşează pentru că nu există resursele necesare.

O tranziție declanşabilă la apariția evenimentului asociat se numeşte *validată.* In *Fig. 5.2* este validată T1 iar în Fig. 5.3 este validată T2. In acest din urmă caz dacă se apasă butonul de pornire *p* se declanşează T2 şi devine activă P3. Motorul este pornit. Acum în P3 se găseşte o marcă şi T3 este validată. La apăsarea butonului de oprire *o* se declanşează T3 şi se ajunge din nou în situația din Fig. 5.3 cu P2 activă şi motorul oprit. Se observă că la apăsarea butoanelor de pornire şi de oprire pozițiile P2 şi P3 devin pe rând active şi se formează un ciclu repetitiv. Poziția P1 nu mai poate devenii nici odată activă, adică alimentarea automatului şi a motorului nu poate fi deconectată. Această situație descoperită în urma analizei evoluției sistemului cu evenimente discrete SED descris de rețeaua Petri din Fig. 5.3 este inacceptabilă şi trebuie remediată.

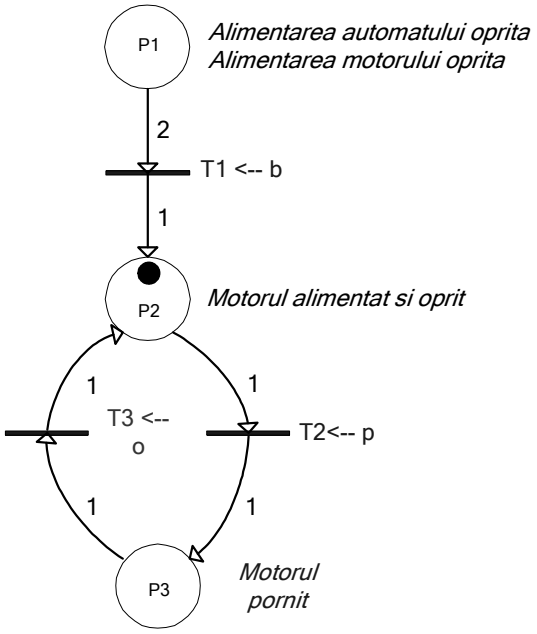


Fig. 5.3 Rețeaua Petri interpretată din Fig. 5.2 *după acționarea butonului de alimentare. Poziția P2 este activă şi se execută acțiunea: motorul alimentat şi oprit.*

Un fenomen interesant apare atunci când se apasă simultan pe butoanele de pornire şi oprire iar poziția P2 sau P3 este activă. Sistemul va bascula. Dacă

motorul este oprit va porni şi dacă este pornit se va opri. Acest fenomen de basculare este în general inacceptabil şi se înlătură. Pentru SED complexe modelarea cu ajutorul rețelelor Petri este foarte utilă deoarece permite descoperirea şi înlăturarea diferitor defecțiuni sau fenomene nedorite.

Rețeaua Petri RP sub formă grafică este suficientă în cazurile simple. Pentru SED complexe se face analiza cu ajutorul calculatorului şi aceasta implică existența unui model matematic format din următorul 5-uplu[58]

$$RP(N, M_0) = \langle P, T; \mathrm{Pr}e, Post; M_0 \rangle \tag{5.1}$$

$$N = (P, T; \mathrm{Pr}e, Post) \tag{5.2}$$

în care N desemnează *structura rețelei,*

M_0: vectorul marcaj inițial al rețelei.

P: mulțimea pozițiilor,

T: mulțimea tranzițiilor,

Pre: matricea de incidență precedentă (înapoi) a tranzițiilor,

Post: matricea de incidență posterioară (înainte) a tranzițiilor,

Pentru rețeaua Petri din *Fig. 5.2* se obține:

$$P = \{P1, P2, P3\} \tag{5.3}$$

$$T = \{T1, T2, T3\} \tag{5.4}$$

Matricele Pre şi Post depind de *structura N* a rețelei. Matricea Pre se construieşte cu ajutorul tabelului următor în care se trec valorile arcelor care leagă fiecare tranziție cu poziția precedentă corespunzătoare. De exemplu, înainte de tranziția T1 se găseşte poziția P1 de care este legată cu un arc orientat cu valoarea 2.

P/t	**T1**	**T2**	**T3**
P1	2	0	0
P2	0	1	0
P3	0	0	1

Deci:

$$\mathrm{Pr}e(p,t) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

[58] Cvintuplu.

In mod asemănător rezultă matricea Post considerând valorile arcelor care leagă fiecare tranziție cu o poziție posterioară.

$$Post(p,t) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{5.6}$$

Vectorul marcajelor inițiale M_0 este:

$$M_0(p) = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \tag{5.7}$$

Rețeaua Petri $RP(N,M_0)$ poate fi caracterizată cu ajutorul următoarei matrici de incidență C(p,t):

$$\forall p \in P, \quad \forall t \in T: \qquad C(p,t) = Post(p,t) - \Pr e(p,t) \tag{5.8}$$

Dacă se cunoaşte matricea de incidență C(p,t) atunci matricele de incidență înapoi Pre(p,t) şi înainte Post(p,t) pot fi calculate cu ajutorul următoarelor relații:

$$Post(p,t) = \max[0, C(p,t)] \tag{5.9}$$

$$\Pr e(p,t) = \max[0, -C(p,t)] \tag{5.10}$$

Pentru rețeaua Petri interpretată din *Fig. 5.2* matricea de incidență *C* rezultă din (5.8), (5.5) şi (5.6)

$$C(p,t) = Post(p,t) - \Pr e(p,t) = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} - \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

(5.11)

Matricea de incidență C depinde de *structura N* a rețelei Petri $RP(N,M_0)$.

5.2. Definirea sistemelor discrete logice.

Sistemele discrete logice *sunt sisteme cu evenimente discrete* care pot fi modelate cu ajutorul unor rețele Petri *interpretate şi sigure*.

Interpretarea rețelei Petri a fost explicată pe larg în paragraful precedent. Condiția de interpretare impune ca pe lângă variabilele de stare ale sistemului, exprimate prin vectorul său de marcaj M, să existe şi variabile *logice* de intrare şi ieşire (Fig. 5.1). Odată cu intrările care interpretează tranzițiile se introduce şi

variabila independentă *timp.* Deoarece o intrare, sub forma unui eveniment, influenţează comportarea sistemului numai dacă tranziţia pe care o interpretează este validată, se observă că timpul intervine numai prin valoarea sa relativă Δt parcursă de la activarea poziţiei conectată la tranziţie. Comportarea dinamică a sistemelor discrete logice este determinată de producerea (apariţia) unor *evenimente*[59] la momente discrete asincrone de timp şi nu este *antrenată de timp*[60] ca în cazul sistemelor continue sau discrete cu eşantionare.

O reţea Petri este *sigură* dacă în toate situaţiile de funcţionare a sistemului cu evenimente pe care îl modelează poziţiile au un număr de mărci mai mic sau egal cu unu. Asupra siguranţei reţelelor Petri se va reveni în paragrafele următoare în legătură cu proprietatea de mărginire a sistemelor cu evenimente discrete. Proprietatea de siguranţă este asemănătoare cu stabilitatea sistemelor continue sau discrete cu eşantionare. Ea impune de asemenea ca variabilele de stare ale sistemului să fie de tip *logic,* la fel cu intrările şi ieşirile sistemului*,* şi în felul acesta sistemul să poată fi implementat cu ajutorul unor dispozitive specifice, cum ar fi contactele şi releele sau automatele programabile logice.

5.3. Tipuri de reţele Petri.

Reţelele Petri sunt de tipuri variate şi îşi găsesc aplicaţii în diferite domenii ale ştiinţei şi ingineriei. Prezentăm pe scurt în continuare tipurile de reţele Petri folosite în această lucrare[61].

5.3.1. Reţele Petri autonome[62]*.*

Acestea sunt reţelele Petri clasice numite adeseori simplu reţele Petri şi sunt definite de (5.1) şi (5.2). Poziţiile şi tranziţiile acestor reţele nu sunt interpretate, deci nu există intrări şi ieşiri. Sistemul cu evenimente discrete modelat de o reţea Petri autonomă este caracterizat numai de variabilele sale de stare exprimate prin vectorul de marcaj M. Tranziţiile aceste reţele se declanşează secvenţial şi instantaneu de fiecare dată când sunt validate. Nu există două tranziţii declanşate simultan (sau în paralel). Mărcile pot rămâne în poziţii orice durată de timp. Acest tip de reţea este utilizat numai pentru studiul proprietăţilor structurale, care nu depind de timp, ale sistemelor cu evenimente discrete şi ale sistemelor discrete logice. Trecerea de la un marcaj la altul se face pas cu pas, declanşând secvenţial tranziţiile.

5.3.2. Reţele Petri interpretate.

Aceste reţele au fost prezentate în paragraful 5.1. Tranziţiile şi poziţiile sunt interpretate prin ataşarea unor evenimente şi acţiuni care reprezintă semnalele logice de intrare şi ieşire ale sistemului. Ori de câte ori o poziţie este activă acţiunile ataşate ei se execută. Evenimentul logic ataşat tranziţiei o interpretează în sensul că tranziţia se declanşează numai dacă evenimentul a apărut, adică ia

---

[59] Sistemele discrete logice sunt de tip *event – driven.*

[60] Sistemele continue sau discrete cu eşantionare sunt de tip *time – driven.*

[61] Alte tipuri de reţele Petri sunt reţelele Petri stochastice, reţelele Petri colorate şi altele.

[62] Se mai numeşte reţea Petri netemporizată sau reţea Petri poziţii – tranziţii.

valoarea logică 1. Evenimentul care interpretează tranziţia poate fi o funcţie logică de alte evenimente externe sistemului sau poate fi un eveniment particular, numit evenimentul prezent întotdeauna şi notat cu e. Reţelele Petri interpretate sigure modelează un tip particular de sisteme cu evenimente discrete numite sisteme discrete logice.

5.3.3. Reţele Petri temporizate.

Temporizarea unei tranziţii se face prin interpretarea ei cu un eveniment special care apare după un interval determinat de timp de la validarea ei. O reţea Petri temporizată are toate tranziţiile temporizate. Dacă temporizarea tuturor tranziţiilor se face cu un interval de timp de durată zero se obţine o reţea Petri autonomă. În *Fig. 5.6* este prezentat un exemplu de reţea Petri temporizată.

5.4. Analiza structurală a sistemelor cu evenimente discrete.

Un rol foarte important în comportarea unui sistem cu evenimente discrete îl au proprietăţile lui structurale puse în evidenţă de reţeaua Petri care îl modelează. Aceste proprietăţi depind de N din (5.2).

5.4.1. Structuri tip folosite la modelarea cu reţele Petri

Cele mai importante structuri care apar într-o reţea Petri sunt: secvenţa (*Fig. 5.10*a), alegerea *(Fig. 5.11* a), saltul (*Fig. 5.10* b), repetarea (*Fig. 5.10* c), convergenţa (*Fig. 5.12*a), paralelismul (*Fig. 5.13* a) şi sincronizarea (*Fig. 5.14* a).

Secvenţa (*Fig. 5.10*a) reprezintă o succesiune liniară de poziţii care sunt activate una după alta prin declanşarea tranziţiilor.

Alegerea (*Fig. 5.11* a) permite trecerea de la o poziţie activă în amonte la una dintre alte poziţii active situate în aval. Existenţa unei structuri tip alegere implică întotdeauna un conflict structural sau efectiv, studiat în 5.4.7, care poate fi eliminat prin interblocare 5.4.8

Saltul (*Fig. 5.10* b)are loc într-o secvenţă liniară atunci când se doreşte evitarea activării unei poziţii. Saltul implică şi o alegere, astfel încât trebuiesc rezolvate toate problemele legate de conflictul tranziţiilor care apare la această structură.

Repetarea (*Fig. 5.10* c) este şi ea o alegere particulară care permite reactivarea unei poziţii dintr-o secvenţă. Această structură conduce la o reţea Petri impură (degenerată) şi trebuie transformată sau evitată aşa cum se arată în 5.4.3.

Convergenţa (*Fig. 5.12* a) este structura duală alegerii şi este folosită atunci când se doreşte ca două sau mai multe acţiuni să se termine în aceiaşi manieră.

Paralelismul structural[63] (*Fig. 5.13* a), contrar alegerii care stabileşte o singură acţiune care se poate realiza, permite mai multor activităţi independente să se deruleze în paralel.

Sincronizarea (Fig. 5.14 a) permite aşteptarea sfârşitului mai multor acţiuni care se realizează în paralel pentru continuarea lor printr-o aceiaşi altă acţiune

[63] Există la reţeaua Petri şi un paralelism tip conveier studiat în 5.7. Grafcetul, o reţea Petri Particulară, permite un paralelism realizat cu ajutorul structurii tip alegere 5.9.6.

5.4.2. Rețele Petri ordinare.

O rețea Petri se numeşte ordinară dacă ponderea tuturor arcelor sale este egală cu unu. Matricele Pre şi Post au toate elementele egale cu 1. Un exemplu se prezintă în *Fig. 5.5*. Se observă că ponderea unitară a arcelor nu mai este specificată.

5.4.3. Rețele Petri pure.

Rețeaua Petri este *pură* sau *nedegenerată* dacă nu are *bucle elementare* formate dintr-o singură poziție şi o singură tranziție. In acest caz se asigură independența între consumarea şi producerea mărcilor (resurselor) în momentul declanşării tranzițiilor. Bucla din *Fig. 5.5* este formată din poziția P1 şi tranziția T4. Condiția de puritate este:

$$\forall p,t: \qquad \Pr e(p,t).Post(p,t) = 0 \tag{5.12}$$

Buclele elementare care impurifică rețeaua sunt, în general, inutile pentru modelarea sistemelor discrete logice. Multe dintre rezultatele teoretice obținute în cadrul analizei rețelelor Petri sunt valabile numai pentru rețelele pure. Din această cauză este recomandabil ca buclele elementare să fie eliminate sau transformate. Se observă că o buclă elementară introduce şi un conflict structural 5.4.7. Eliminarea acestuia se poate face prin interblocare, ca în paragraful 5.4.8, iar bucla elementară devine de prisos putând fi înlăturată. Transformarea buclei elementare se realizează introducând suplimentar în buclă o poziție si o tranziție fictive. In felul acesta rețeaua Petri nu mai este degenerată.

5.4.4. Poziția sursă sau receptor.

O poziție fără nici o tranziție de intrare se numeşte *poziție sursă*, de exemplu P_1 în *Fig. 5.2*. Poziția fără nici o tranziție de ieşire este o *poziție receptor*.

5.4.5. Tranziția validată[64].

Condiția de *validare* a unei tranziții t_1, care asigură existența resurselor ce urmează să fie consumate la declanşarea tranziției, se exprimă astfel:

$$t_1 \in T, \quad \forall p \in P: \qquad M(p) \geq \Pr e(p,t_1) \tag{5.13}$$

Cu alte cuvinte, numărul mărcilor existente în pozițiile precedente tranziției este suficient pentru a fi consumate la declanşarea sa.

De exemplu, pentru rețeaua Petri din *Fig. 5.2*

$$M_0(p) \geq \Pr e(p,T_1) \Rightarrow \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \geq \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \tag{5.14}$$

şi deci tranziția T_1 este validată pentru marcajul inițial M_0.

[64] Enabled (eng.)

5.4.6. Tranziția declanşată[65].

O tranziție este declanşată dacă este validată şi evenimentul care o interpretează a apărut (are valoarea logică unu). Prin declanşare se îndepărtează un număr de mărci din fiecare poziție de intrare şi se adaugă un număr de mărci în fiecare poziție de ieşire. Numărul mărcilor îndepărtate, respectiv adăugate, este determinat de ponderea arcelor de intrare, respectiv ieşire, din tranziție.

5.4.7. Conflictul structural şi conflictul efectiv al tranzițiilor.

Două tranziții sunt în *conflict* structural dacă au cel puțin o poziție în comun la intrare. Aceleaşi tranziții sunt în plus şi în conflict în *conflict efectiv* pentru un marcaj M dacă amândouă sunt declanşabile şi resursele sunt insuficiente. Tranzițiile T2 şi T4 din *Fig. 5.5* sunt în conflict structural şi efectiv. Condiția de conflict structural a tranzițiilor oarecare t_1 şi t_2 este:

$$\exists p: \qquad \Pr e(p,t_1).\Pr e(p,t_2) \neq 0 \tag{5.15}$$

Tranzițiile T_2 şi T_4 din *Fig. 5.5* sunt în conflict structural deoarece:

$$P_2: \qquad \Pr e(P_2,T_2).\Pr e(P_2,T_4) = 1 \tag{5.16}$$

Aceleaşi două tranziții t_1 şi t_2 sunt în plus şi în conflict efectiv pentru un marcaj M(p) dacă amândouă sunt declanşabile şi resursele sunt insuficiente:

$$M(p) < \Pr e(p,t_1) + \Pr e(p,t_2) \tag{5.17}$$

Dacă poziția P_2 este activă fiecare dintre tranzițiile T_2 şi T_4 este validată şi deci declanşabilă. Condiția (5.17) este verificată pentru poziția B:

$$M(P_2) < \Pr e(P_2,T_2) + \Pr e(P_2,T_4) = 2 \tag{5.18}$$

şi deci tranzițiile T_2 şi T_4 sunt în conflict efectiv. Aceasta înseamnă că dacă motorul este oprit în P_2 şi se apasă simultan butoanele de pornire p ăi oprire o, amândouă tranzițiile T_2 şi T_4 sunt declanşate ducând la două situații complet diferite şi incompatibile: în P_3 motorul este pornit iar în P_2 motorul este oprit. Deoarece regulile de funcționare ale rețelei Petri interpretate impun ca la un anumit moment de timp să poată fi declanşată numai o singură tranziție, rezultă că în cazul studiat se va declanşa T_2 sau T_4 în funcție de factori aleatori care fac ca unul dintre evenimentele care le interpretează să apară cu o fracțiune de secundă mai devreme.

Distincția dintre conflictele structurale şi efective are sens numai pentru rețelele Petri interpretate sau temporizate. În cazul unei rețele Petri autonome existența unui conflict structural conduce la o comportare nedeterministă a sistemului pe care îl modelează.

[65] Fired (eng).

5.4.8. Interblocarea prin interpretare.

Eliminarea conflictului efectiv între două tranziţii se poate face condiţionând astfel evenimentele care le interpretează încât să nu poată apărea simultan. Această acţiune se numeşte *interblocare.* Pentru reţeaua din *Fig. 5.5* s-a impus condiţia ca tranziţia T_2 să fie interpretată de un eveniment care apare, sau ia valoarea logică unu, numai atunci când se apasă pe butonul de pornire p şi *nu* se apasă butonul de oprire o.

$$T_2 \leftarrow p.\bar{o} \qquad (5.19)$$

Prin interblocare se pot înlătura şi buclele care fac reţeaua degenerată (impură). De exemplu, interpretarea tranziţiei T_2 din (5.19) face inutila prezenţa buclei care începe şi se termină în P_2.

5.5. Analiza comportamentală a sistemelor cu evenimente discrete.

5.5.1. Ecuaţia de stare.

Declanşarea diferitor tranziţii[66] a reţelei Petri interpretate face ca starea sistemului cu evenimente discrete SED pe care îl modelează, descrisă de marcajul său M(p), să se modifice. De exemplu, în *Fig. 5.2* starea este dată de M_0 definit de (5.7). Declanşarea tranziţiei T_1 conduce la starea din Fig. 5.4 cu marcajul M_1 din (5.20). Asemănător declanşarea lui T_2 provoacă apariţia marcajului M_2 cu valorile date de (5.20). Se poate spune că o secvenţă de declanşări de tranziţii *s* definită de (5.21) conduce sistemul cu evenimente discrete din starea M_0 în starea *M*.

$$T_1 \Rightarrow \quad M_1(p) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \qquad T_2 \Rightarrow \quad M_2(p) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (5.20)$$

Pentru secvenţa *s* (5.21) se defineşte *vectorul caracteristic* s^* corespunzător (5.22) definit ca vectorul[67] cu indici în *t,* definit de (5.22), şi componente care reprezintă numărul de apariţii a fiecărei tranziţii în secvenţa *s.*

$$s = T_1 T_2 T_3 T_2 \qquad (5.21)$$

t	**s^*(t)**
T_1	*1*
T_2	*2*
T_3	*1*

[66] Reamintim că tranziţia se declanşează dacă este validată şi evenimentul care o interpretează a apărut.

[67] Acesta mai este numit vectorul de numărare a execuţiilor.

$$s^* = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \qquad (5.22)$$

Legătura dintre două marcaje a unei rețele Petri interpretate *pure* este dată de *ecuația fundamentală de stare*:

$$M = M_0 + C.s^* \qquad (5.23)$$

în care C este matricea de incidență definită de (5.8).

Această ecuație permite calculul *stării* unui sistem cu evenimente discrete (marcajul M) dacă se cunoaşte starea inițială M_0 a sistemului, *funcția de tranziție* a stărilor (matricea de incidență C) şi *intrarea* sub forma unei secvențe de tranziții *declanşabile s* determinată de *vectorul caracteristic* s^*. Folosind ecuația fundamentală sistemele cu evenimente discrete pot fi analizate cu ajutorul metodelor algebrei liniare, în mod asemănător cu analiza sistemelor continue.

Pentru exemplul considerat până acum referitor la rețeaua din *Fig. 5.2* cu matricea de incidență (5.11) şi secvența (5.21) rezultă noua stare:

$$M = M_0 + C.s^* = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (5.24)$$

Trebuie subliniat că ecuația fundamentală (5.23) se referă numai la rețelele Petri interpretate *pure* şi este valabilă numai pentru secvențe de tranziții *declanşabile*, adică de tranziții care îndeplinesc, fiecare separat, condiția de *validare* (5.8). Matricea de incidență C depinde numai de structura N a rețelei definită de (5.2).

5.5.2. Graful marcajelor accesibile.

Graful marcajelor accesibile al unei rețele Petri RP=(N,M_0) este graful care are drept vârfuri mulțimea marcajelor M_i care se obțin plecând din M_0 şi drept arce tranzițiile corespunzătoare.

5.6. Performanțele sistemelor cu evenimente discrete.

Performanțele SED sunt determinate de anumite valori pe care o iau mărimile de ieşire. Aceste mărimi sunt la rândul lor sunt determinate de starea SED, adică de marcajul său M.

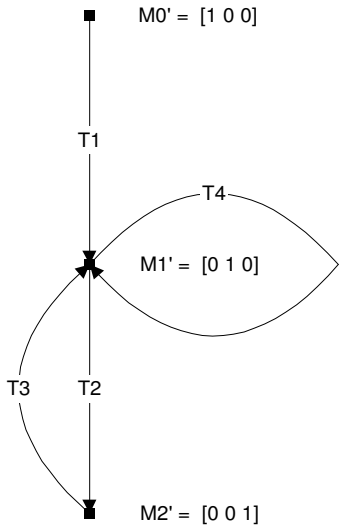


Fig. 5.4 Graful marcajelor accesibile M' pentru reţeaua Petri din Fig. 5.5

5.6.1. Reversibilitatea[68]

Un interes deosebit, din punct de vedere practic, îl prezintă secvenţele de intrare *s* pentru care starea finală a sistemului cu evenimente discrete este egală cu starea sa iniţială, adică:

$$M = M_0 \qquad (5.25)$$

Din relaţiile (5.23) şi (5.25) rezultă condiţia de *reversibilitate staţionară* a unei secvenţe s^*

$$C.s^* = 0 \qquad (5.26)$$

Tranziţiile care formează secvenţa care îndeplineşte condiţia (5.26) formează o componentă reversibilă staţionar a reţelei Petri interpretate. Dacă această componentă cuprinde *toate* tranziţiile reţelei Petri staţionare atunci reţeaua se numeşte reversibilă staţionar. Sistemele cu evenimente discrete *automate* trebuie să fie caracterizate de reţele Petri reversibile staţionar. In felul acesta se asigură repetabilitatea anumitor activităţi.

[68] Reversibility (eng.)

Dacă se aplică relaţia (5.26) la reţeaua din *Fig. 5.2* se obţine:

$$Cs^* = \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} . \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = 0 \qquad (5.27)$$

Soluţia acestui sistem de ecuaţii va fi $s_1=0$ şi $s_2=s_3$. Deci un vector caracteristic care satisface condiţia (5.26) este:

$$s^* = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \qquad (5.28)$$

Componenta reversibilă a reţelei va fi compusă din tranziţiile T_2 şi T_3 după cum rezultă şi din *Fig. 5.2*.

5.6.2. Mărginirea şi siguranţa.

O reţea Petri este *mărginită*[69] dacă numărul de mărci din fiecare poziţie are o valoare mărginită pentru toate situaţiile de funcţionare. Vectorul caracteristic *s** al secvenţei de intrare *s* definit de (5.22) este un model matematic care condenseaza şi reduce informaţia referitoare la intrarea sistemului cu evenimente discrete. Un pas mai departe în această direcţie constă în efectuarea produsului scalar între vectorul marcaj şi un vector *f* care ponderează numărul de mărci din fiecare poziţie a reţelei. Ecuaţia fundamentală (5.23) va deveni atunci:

[69] Bounded (eng.)

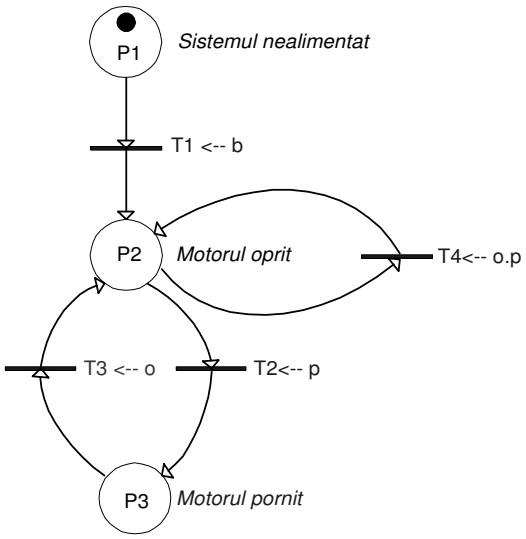


Fig. 5.5 O rețea Petri interpretată ordinară, sigură, degenerată, neviabilă, cu conflict structural şi efectiv între T_2 şi T_4 .

$$f^T.M = f^T.M_0 + f^T.C.s^* \tag{5.29}$$

în care f^T este transpusul vectorului f.

Se demonstrează că rețeaua este mărginită dacă există un vector *f* pozitiv astfel încât să fie îndeplinită condiția:

$$\exists f > 0, \quad f^T C \le 0 \tag{5.30}$$

Dacă se aplică această relație la rețeaua din *Fig. 5.2* se obține:

$$f^T C = \begin{bmatrix} f_A & f_B & f_C \end{bmatrix} \cdot \begin{bmatrix} -2 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} (-2f_A + f_B) & (-f_B + f_C) & (f_B - f_C) \end{bmatrix} = 0 \tag{5.31}$$

Soluția acestui sistem de ecuații liniare este:

$$2f_A = f_B = f_C \tag{5.32}$$

Deci există un vector *f* pozitiv şi rețeaua este mărginită.

Rețeaua Petri este denumită *sigură*[70] sau *binară sau logică* pentru un anumit marcaj M_0 dacă toate pozițiile au un număr de mărci egal sau mai mic decât unu pentru toate situațiile de funcționare apărute plecând de la acest marcaj inițial. In *Fig. 5.5* se prezintă o rețea Petri interpretată sigură. Deoarece pozițiile pot avea maximum o marcă, arcele orientate nu mai sunt evaluate, aşa cum se întâmplă în *Fig. 5.5*. Rețeaua Petri este în acest caz ordinară. Rețelele Petri sigure modelează numai operațiile din sistem nu şi distribuirea resurselor.

Performanța de mărginire este asemănătoare cu performanța de stabilitate a sistemelor continue. În ambele cazuri ieşirea sistemului trebuie să fie finită.

5.6.3. Viabilitatea.

O rețea Petri RP=(N,M_0) este *viabilă*[71] dacă pentru orice marcaj M realizat pornind din M_0 se poate declanşa în continuare orice tranziție a rețelei N, eventual după declanşarea unui număr finit de alte tranziții. Dacă se ajunge la un marcaj pentru care nici o tranziție a rețelei nu mai poate fi executată, acesta se numeşte *blocaj*[72].

Performanțele de reversibilitate, mărginire şi viabilitate joacă un rol important în studiul rețelelor Petri. Aceste performanțe nu depind una de alta.

[70] Safe (eng.)
[71] Live (eng.)
[72] Deadlock (eng.)

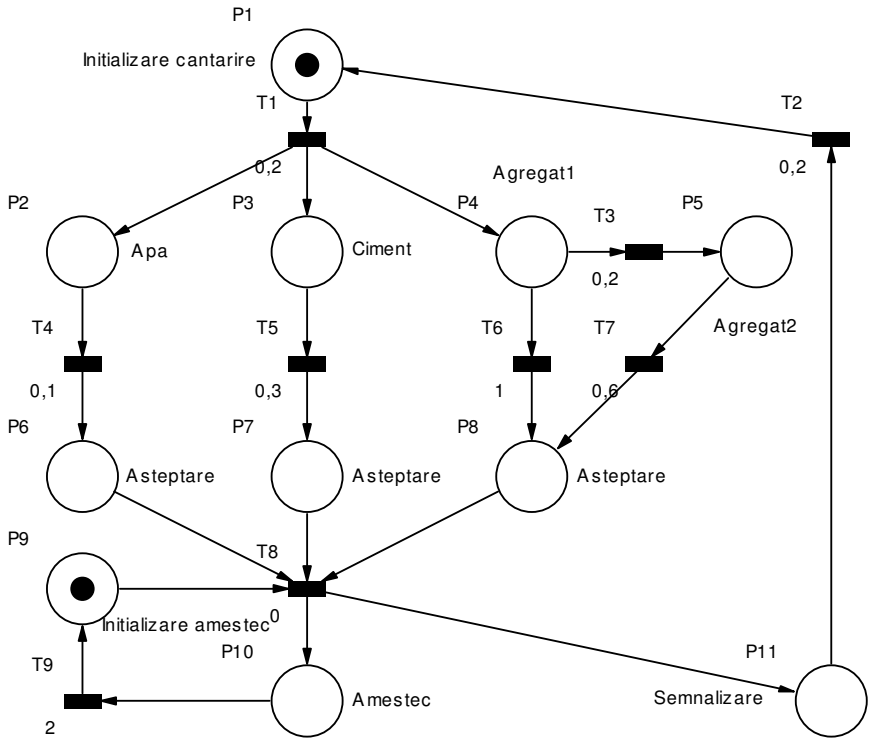


Fig. 5.6 Modelul automatului unei staţii de betoane sub forma unei reţele Petri temporizate şi sigure.

5.7. Sisteme cu evenimente discrete şi evoluţie paralelă.

Un exemplu de sistem discret logic va permite explicarea conceptului de paralelism.

Automatul care conduce o staţie de betoane este modelat cu ajutorul unei reţele Petri temporizate sigure în *Fig. 5.6*. El trebuie să asigure cântărirea apei, cimentului, agregatelor şi să comande amestecul lor.

Fiecare poziţie şi tranziţie a reţelei Petri are un nume şi este interpretată. Toate tranziţiile au ataşate evenimente[73] sub forma unor temporizări. De exemplu, evenimentul tranziţiei T1, specificat în dreptul ei, apare după $\tau = 0{,}2$ minute de la validarea[74] sa. Reţelele Petri de acest tip se numesc *temporizate*. Poziţiile P1 şi P9 sunt active în momentul iniţial şi atunci când cântarele şi malaxorul sunt goale. Să vedem dacă automatul are performanţa de reversibilitate, adică revine în starea iniţială după o secvenţă finită de operaţiuni. În situaţia din *Fig. 5.6* este validată numai tranziţia T1 care se declanşează atunci când apare evenimentul de temporizare care o interpretează, adică după 0,2 minute. După declanşarea sa

[73] Sunt interpretate într-un mod specific.

[74] O tranziţie este validată dacă toate poziţiile precedente sunt active.

poziția P1 devine inactivă iar pozițiile P2, P3, P4 sunt active şi se execută acțiunile care le interpretează, adică se umplu recipientele de apă, ciment şi agregat1. Cântărirea se face prin temporizare, adică, de exemplu, cantitatea dorită de apă se obține după o umplere a rezervorului de 0,1 minute. Asemănător, pentru un debit constant de umplere, cantitatea dorită de ciment se obține după 0,3 minute.

În cazul poziției P4 avem un conflict structural deoarece atunci când este activă sunt validate tranzițiile T6 şi T3. Conflictul nu este efectiv însă deoarece evenimentele care interpretează tranzițiile nu apar simultan, având temporizări diferite de 1 şi 0,2 minute. Deci, după activarea simultană a pozițiilor P2,P3 şi P4 devin active pe rând P6, P5, P7 şi P8. În P5 se cântăreşte agregatul cu numărul 2 iar în restul pozițiilor se aşteaptă terminarea cântăririi tuturor componentelor. Cântărirea agregatelor se face prin acumulare în acelaşi recipient.

Se observă că operațiile de cântărire a apei, cimentului şi agregatelor se execută în paralel, adică în acelaşi timp[75]. Acest tip de execuție simultană se numeşte *paralelism general.*

La terminarea cântăririi componentelor se validează tranziția T8, deoarece P6, P7, P8 şi P9 sunt active şi se declanşează imediat deoarece este interpretată cu o temporizare zero. Pozițiile P10 şi P11 devin active. Acțiunile care le interpretează specifică începerea amestecului componentelor betonului şi semnalizarea terminării cântăririi. Tranzițiile T9 şi T2 sunt validate de se vor declanşa la intervale de timp diferite, respectiv 2 şi 0,2 minute. Deci mai întâi se declanşează P1 care inițializează cântărirea pentru şarja a doua de betoane.

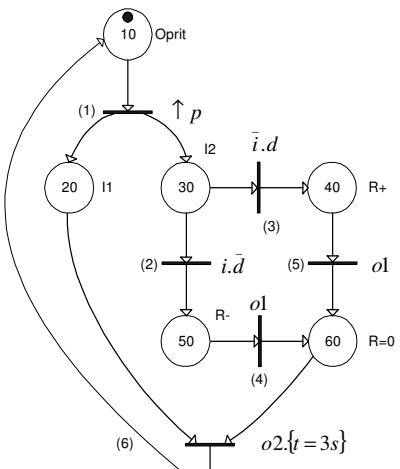
Apa, cimentul şi agregatele şarjei a doua se transportă în recipientele respective şi se cântăresc prin temporizare în timp ce componentele primei şarje se amestecă. Deci cântărire pentru şarja a doua şi amestecul pentru prima şarjă se execută în acelaşi timp, în paralel. Această execuție simultană se numeşte *paralelism conveier*[76].

A doua cântărire se termină şi amestecul primei şarje nu este încă terminat. Se aşteaptă în pozițiile P6, P7 şi P8 ca să se inițializeze un nou amestec în poziția P9. Când aceasta se întâmplă se reiau operațiile simultane de cântărire şi amestec.

---

tipuri

a)



două

Fig. 5.7 Modelul unui sistem discret logic, format din instalațiile I1, I2 şi robinetul R, prezentat sub forma unei rețele Petri interpretate, sigure, viabile şi fără conflicte a) şi sub forma unui grafcet b).

Marcajul inițial format din pozițiile P1 şi P9 active nu se mai obține niciodată. Rețeaua Petri din *Fig. 5.6* nu este *reversibilă.* Pentru evitarea acestui fenomen la terminarea cântăririi numărului dorit de şarje se modifică evenimentul care interpretează tranziția T1 astfel încât aceasta să nu se mai declanşeze automat, cu o temporizare de 0,1 minute, după activarea poziției P1.

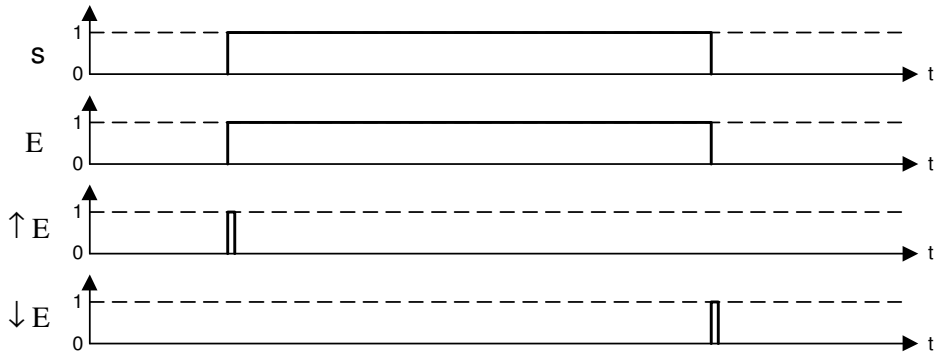


Fig. 5.8 Semnalul logic s de intrare în sistem şi evenimentele tip nivel E şi tip impuls ↑E, ↓E ataşate.

5.8. Maşina de stare.

Maşina de stare[77] este un sistem cu evenimente discrete care poate fi modelat printr-o rețea Petri la care fiecare tranziție este legată la exact o poziție de intrare şi o poziție de ieşire prin arce cu ponderea 1.

$$\forall t \in T \qquad \sum_{p \in P} \Pr e(p,t) = 1 \qquad \sum_{p \in P} Post(p,t) = 1 \qquad (5.33)$$

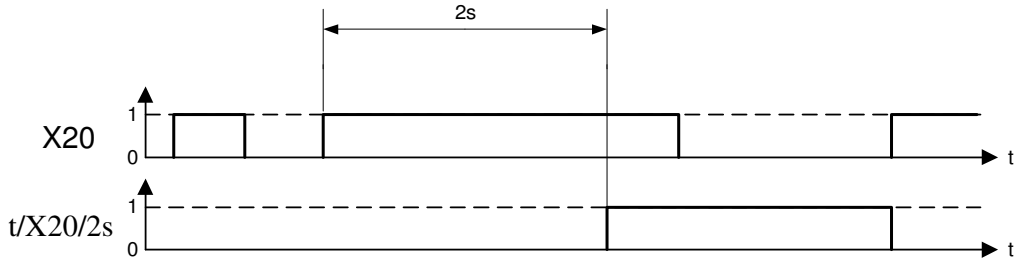


Fig. 5.9 Variabila t/X20/2s de temporizare față de activarea etapei 20

Dacă maşina de stare este sigură, atunci marcajul inițial M_0 conține o singură marcă. Aceasta se va deplasa în cadrul evoluției sistemului din poziție în

[77] State machine (eng.)

poziție indicând starea activă curenta. O maşină de stare interpretată şi sigură mai poartă numele de *automat finit*. In acest caz graful marcajelor accesibile este izomorf cu graful rețelei obținut prin ştergerea tranzițiilor. Reciproc, dacă graful marcajelor accesibile ale unei rețele Petri oarecare este finit i se poate asocia o maşină de stare. Deoarece maşina de stare poate fi implementată relativ uşor rezultă că orice rețea Petri cu graful marcajelor accesibile *finit* poate fi implementată.

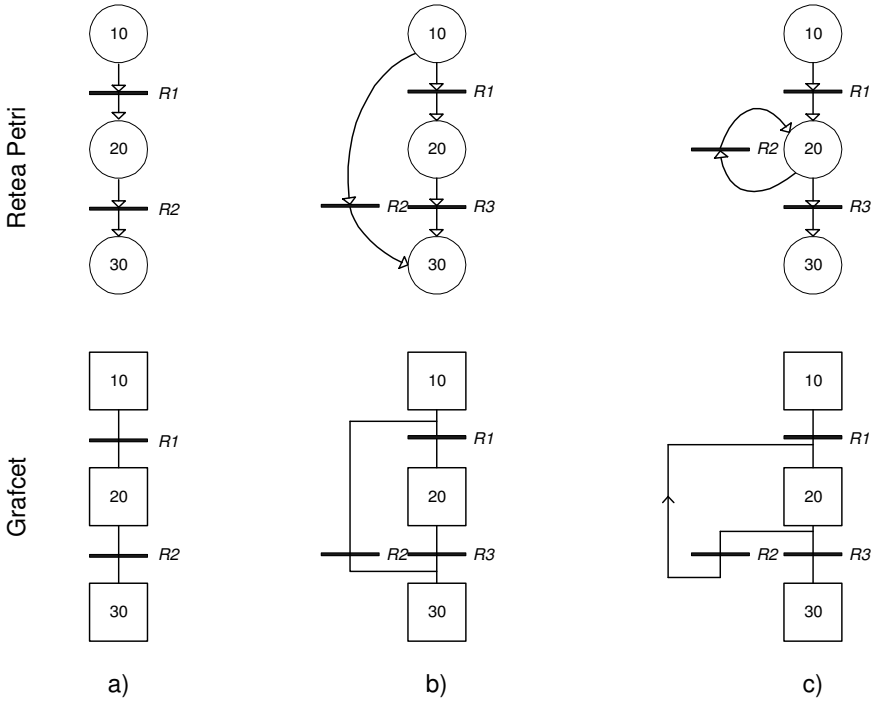


Fig. 5.10 Structurile tip secvență a), salt b) şi repetare c) la rețeaua Petri şi grafcet.

Condiția de *viabilitate* a maşinii de stare este dată de teorema lui Commoner: Condiția necesară şi suficientă pentru ca rețeaua Petri să fie viabilă este ca graful obținut din rețea prin ştergerea tranzițiilor să fie tare conex şi marcajul inițial să conțină cel *puțin* o marcă.

Condiția de *siguranță* pentru o maşină de stare impune ca marcajul inițial M_0 să aibă cel *mult* o marcă.

O maşină de stare *viabilă* este şi *sigură* dacă şi numai dacă marcajul inițial M_0 are o *singură* marcă.

5.9. Grafcetul.

5.9.1. Trecerea de la rețeaua Petri la grafcet.

Rețelele Petri interpretate, sigure, viabile şi fără conflicte efective sunt numite *conforme* şi coincid cu o clasă de grafceturi frecvent utilizate în practică. Cuvântul *grafcet*[78] provine din inițialele GRAF de Comandă Etape - Tranziții – *GRAFCET*[79].

Grafcetul este un model folosit foarte mult pentru sistemele discrete logice, a fost standardizat internațional[80] şi a devenit unul din instrumentele de bază pentru programarea automatelor programabile logice. Trecerea de la o rețea Petri conformă la un grafcet se face imediat, după cum se vede în *Fig. 5.7*, deosebirile constând numai în formalismul grafic[81].

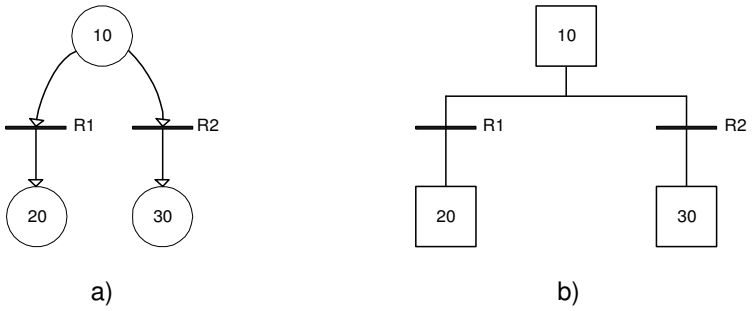


Fig. 5.11 Structura tip alegere la rețeaua Petri a) şi grafcet b).

Exemplul din *Fig. 5.7* se referă la modelul unui sistem discret logic format din două instalații I1, I2 şi un robinet R. La acționarea butonului de pornire *p* ambele instalații pornesc simultan în paralel După aceasta se închide sau se deschide, mai mult sau mai puțin, cu ajutorul unui servomotor reversibil, robinetul R. Pentru aceasta se acționează butonul de închidere *i* sau butonul de deschidere *d.* Oprirea închiderii sau deschiderii robinetului se face cu butonul *o1.* Oprirea tuturor instalațiilor se face cu butonul *o2* şi sistemul ajunge în starea inițială modelată de poziția 10 la rețeaua Petri sau etapa[82] 10 la grafcet. Se observă că pozițiile sunt reprezentate prin cercuri iar etapele grafcet prin pătrate.

---

[78] Prin proveniența sa *grafcet* este un substantiv românesc, deşi coincide cu cuvântul francez. Din această cauză, la fel ca în limba franceză, el se *pronunță* şi se *foloseşte* după regulile limbii române.

[79] Grafcet – Graphe de Commande Etape – Transition (fr.).
SFC – Sequential Function Chart (eng.).

[80] IEC 848 (Function Chart for Control System – FCCS), 1987.

[81] Trecere inversă, de la grafcet la o rețea Petri, nu este directă deoarece grafcetul posedă facilități suplimentare de modelare, cum ar fi macro-etapa, care nu se regăsesc la rețeaua Petri.

[82] Step (eng.)

Tranziția (1) realizează o *distribuție ŞI* iar tranziția (6) realizează o *joncțiune ŞI*. Din etapa 30 urmează o *distribuție SAU* iar etapa 60 realizează o *joncțiune SAU*. Se observă în *Fig. 5.7* modul diferit în care rețeaua Petri şi grafcetul prezintă distribuțiile şi joncțiunile tip SI şi SAU.

Tranzițiile (2) şi (3) sunt în conflict structural pentru rețeaua Petri. Conflictul nu este însă efectiv deoarece s-a realizat o interblocare între acționarea butoanelor *i* şi *d*. Noțiune de conflict structural şi efectiv nu există la grafcet. Dacă n-ar fi existat interblocarea, etapele 40 şi 50 ar fi putut fi activate simultan.

5.9.2. Etape, tranziții şi legături orientate.

Etapa din grafcet corespunde poziției (locației) din rețeaua Petri şi modelează o situație în care comportamentul sistemului modelat, sau numai a unei părți ale sale, rămâne neschimbat în raport cu intrările şi ieşirile sale.

Etapa inițială, activă la momentul de timp inițial, este specificată la grafcet printr-un pătrat dublu. Pentru grafcetul din *Fig. 5.7* b) etapa inițială are numărul 10 iar etapele active la momentul de timp curent sunt specificate prin plasarea unei mărci în pătratele corespunzătoare cu numărul 20 şi 40. Aceasta înseamnă că grafcetul din figură prezintă situația sistemului discret logic în care funcționează instalațiile I1, I2 şi robinetul R se deschide. La rețeaua Petri nu se specifică starea inițială iar pentru cazul din *Fig. 5.7* a) sistemul modelat se găseşte în starea de la momentul de timp curent specificată prin plasarea unei mărci în poziția corespunzătore cu numărul 10.

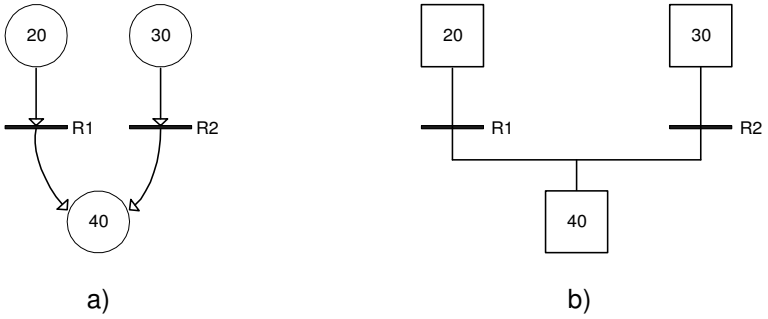


Fig. 5.12 Structura tip convergență la rețeaua Petri a) şi grafcet b).

Acțiunile care interpretează etapele grafcetului sunt trecut în dreptunghiuri ataşate de acestea. Pentru etapele inițiale se consideră că acțiunile implicite sunt cele care opresc toate echipamentele şi se specifică, dacă este cazul, numai acțiunile diferite de acestea.

Fiecărei etape i se asociază o variabilă de stare logică notată X_i, în care i este numărul etapei. Această variabilă ia valoarea logică 1 dacă etapa este activă sau 0 dacă etapa este inactivă.

Tranzițiile şi evenimentele care le interpretează sunt reprezentate la fel în cazul rețelei Petri şi a grafcetului. Denumirea tranzițiilor este formată de un număr plasat între paranteze rotunde. De exemplu, tranziția (5) este interpretată de evenimentul *o1,* o variabilă logică care ia valoarea logică 1 (adevărat) dacă se acționează butonul de oprire cu numărul 1. La grafcet tranzițiile sunt formate din

linii *orizontale* îngroşate, pe când reţelele Petri reprezintă tranziţiile prin dreptunghiuri[83] înnegrite care pot fi situate oricum.

Arcele orientate sunt formate la grafcet numai din *linii drepte şi verticale* cu sensul de sus în jos. Din această cauză săgeţile nu mai sunt prezente. Excepţie fac doar arcele orientate de întoarcere la starea iniţială. La reţeaua Petri, *Fig. 5.7* a), arcele orientate pot avea orice formă, pot fi orientate oricum şi sensul este indicat prin săgeţi.

5.9.3. Interpretarea tranziţiilor.

La grafcet fiecărei tranziţii îi este asociată o receptivitate R_i, în care i este numărul tranziţiei. Receptivitatea este o funcţie logică de evenimente, stări ale etapelor şi temporizări.

Evenimentele sunt, ca şi la reţeaua Petri, variabile logice ataşate semnalelor de intrare. In *Fig. 5.8* se observă că pentru un semnal logic s(t) de intrare se pot defini variabilele logice tip nivel E şi tip impuls ↑E, ↓E care apar pe frontul crescător, respectiv descrescător al semnalului E.

Variabilele de stare logică ataşate fiecărei etape sunt notate cu X_i, în care i este numărul etapei. Aceste variabile iau valoarea logică 1 atâta timp cât etapa este activă.

Variabilele de temporizare se definesc faţă de momentul în care o etapă devine activă. In Fig. 5.9 se prezintă variabila de temporizare t/X20/2s care are valoarea logică 1 după o întârziere de 2 secunde de la activarea etapei cu numărul 20. Se observă că temporizarea apare numai dacă etapa este activă mai mult de 2 secunde. In momentul în care etapa se reactivează temporizarea revine la valoarea logică 0.

5.9.4. Interpretarea etapelor.

Fiecărei etape i se poate ataşa o variabilă logică funcţie de starea etapei la care este ataşată, de semnale de intrare şi de o temporizare. De exemplu

$$A20 = X20.a.t \,/\, X20/5s \qquad (5.34)$$

De exemplu, acţiunea A20 ataşată etapei X20 şi definită de (5.34) apare, adică ia valoarea logică 1 dacă etapa cu numărul 20 este activă, adică X20=1, semnalul de intrare a=1 şi temporizarea *t/X20/5s* a fost realizată.

[83] La reţelele Petri generale tranziţiile consumă un număr resurse (mărci) din poziţiile precedente şi produc un alt număr de mărci în poziţiile posterioare. Deci ele au consistenţă şi apar ca nişte dreptunghiuri ca în *Fig. 5.6*. Pentru reţelele Petri sigure, cu maximum o marcă în fiecare poziţie, folosite la modelarea sistemelor discrete logice, se adoptă, în general, o reprezentare apropiată de cea folosită de grafcet: dreptunghiuri înnegrite subţiri.

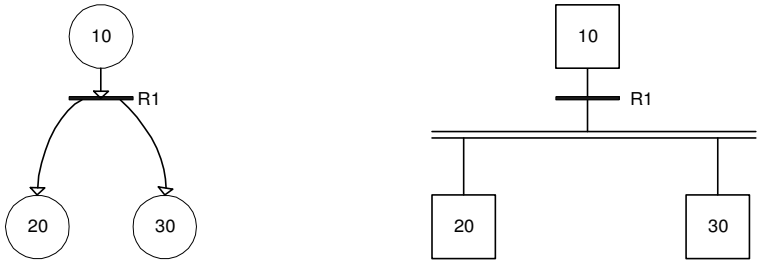


Fig. 5.13 Structura tip paralelism la rețeaua Petri a) şi grafcet b).

5.9.5. Reguli de evoluție în grafcet.

- **Regula 1. Condiții inițiale**. La momentul inițial numai etapele inițiale sunt active.
- **Regula 2. Condiții de validare.** Pentru ca o tranziție să fie validată este necesar ca toate etapele precedente să fie active.
- **Regula 3. Condiții de declanşare.** O tranziție este declanşată dacă este validată şi receptivitatea care o interpretează a apărut. Pentru a declanşa o tranziție trebuie, obligatoriu în această ordine, ca etapele precedente să fie dezactivate iar etapele posterioare să fie activate.
- **Regula 4. Declanşării simultane.** Toate tranzițiile declanşabile la un moment dat sunt declanşate simultan.
- **Regula 5. Conflictul de activare**. Dacă o etapă trebuie simultan activată şi dezactivată de declanşările simultane ale unei tranziții precedente şi ale unei tranziții posterioare, atunci ea rămâne activă[84].

5.9.6. Structuri folosite la modelarea cu grafcet

Structurile tip folosite în grafcet sunt aceleaşi ca la rețeaua Petri 5.4.1: secvența (*Fig. 5.10* a), saltul (*Fig. 5.10* b), repetarea (*Fig. 5.10* c), alegerea (*Fig. 5.11* b) convergența (*Fig. 5.12* b), paralelismul (*Fig. 5.13* b) şi sincronizarea (*Fig. 5.14* b). După cum se vede din aceste figuri şi (*Fig. 5.7*) reprezentarea grafică este diferită.

O problemă apare la structura tip alegere (*Fig. 5.11*). La rețeaua Petri această structură realizează o alegere tip SAU EXCLUSIV iar la grafcet realizează o alegere tip SAU. Deosebirea apare atunci când evenimentele R1 şi R2 apar simultan. La grafcet, dacă etapa 10 este activă, se declanşează tranzițiile pe care aceste evenimente le interpretează si etapele 20 şi 30 devin active simultan, în *paralel*. Rețeaua Petri nu permite aceasta, ca o alegere să se transforme într-un paralelism! Diferența apare din modul diferit de definire a mărcii.

[84] Prin analogie cu pornirea unui motor electric se poate spune că există o prioritate la pornire.

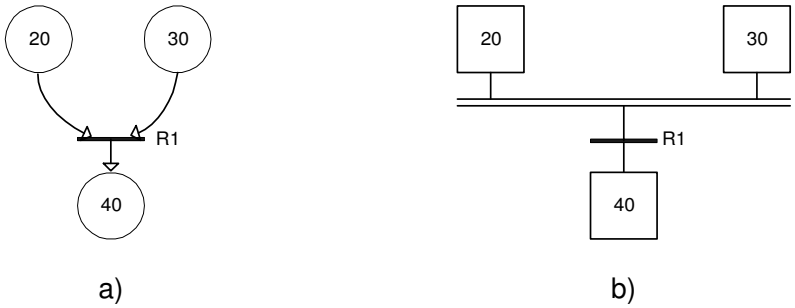


Fig. 5.14 Structura tip sincronizare la rețeaua Petri a) şi grafcet b).

La rețeaua Petri marca este o resursă şi este indivizibilă. Nu se poate deplasa în acelaşi timp din poziția 10 în 20 si 30. Din cauza aceasta apare conflictul efectiv, prezentat în paragraful 5.4.7, care trebuie rezolvat. Dacă nu este rezolvat sistemul discret logic modelat de rețeaua Petri se va comporta nedeterminist, la apariția simultană a evenimentelor R1 şi R2 se va activa etapa 20 sau (exclusiv) etapa 30 în funcție de factori care nu sunt cunoscuți.

La grafcet marca este un indicator, semnalizator al stării de activitate sau inactivitate a etapei. Deoarece mai multe etape pot fi active simultan şi marca se poate deplasa simultan din etapa 10 în etapele 20 şi 30.

5.9.7. Compararea grafcetului cu rețeaua Petri.

● *Avantajele* grafcetului față de rețeaua Petri sunt următoarele:

Grafcetul permite interpretarea tranzițiilor şi în funcție de unele evenimente interne (endogene) cum ar fi, de exemplu, apariția stării *active* sau *inactive*[85] a pozițiilor. Drept urmare desenul, care are o reprezentare grafică standardizată internațional, este mult mai simplu şi mai inteligibil.

Existența macro-etapelor în grafcet conduce de asemenea la o simplificare substanțială a caietelor de sarcini şi programelor pentru automatele programabile logice.

Grafcetul permite declanşarea simultană a tranzițiilor fără să existe noțiunea de conflict. Aceasta uşurează întocmirea şi folosirea caietelor de sarcini de către nespecialişti. O poziție activată şi dezactivată simultan rămâne în grafcet activă[86]. In felul acesta se evită ignorarea acțiunilor corespunzătoare poziției şi modificarea algoritmului care a stat la baza întocmirii grafcetului. Ca un preț pentru facilitățile oferite, grafcetul nu posedă un model matematic atât de elaborat ca rețeaua Petri.

Desenul standardizat al grafcetului este mai adecvat pentru automatele programabile logice.

[85] Rețeaua Petri nu poate testa dacă o poziție este inactivă.
[86] Prioritate la pornire.

● *Dezavantajele* grafcetului faţă de reţeaua Petri sunt următoarele:

Grafcetul permite şi paralelismul prin interpretarea tranziţiilor situate într-o structură tip alegere. În aceste situaţii pot apare frecvent greşeli la ieşirea din paralelism.

La modelarea sistemelor discrete logice cu ajutorul grafcetului pot apare erori la reactivarea etapelor. La modelul reţea Petri aceste erori sunt repede evidenţiate.

Structura modelului grafcet fără interpretare oferă puţine informaţii şi nu permite verificările care se pot face pe un model reţea Petri.

In concluzie se poate aprecia că pentru elaborarea caietelor de sarcini sau programarea automatelor programabile logice grafcetul este preferabil. Pentru analiza, sinteza şi validarea modelului este mai bună reţeaua Petri. La prima vedere ar părea avantajoasă elaborarea caietului de sarcini sub formă de grafcet şi apoi să se treacă la reţeaua Petri pentru analiză, validare şi sinteză. Din păcate trecerea grafcet – reţea Petri şi invers nu se poate efectua direct decât în cazul modelelor mai simple de tip reţea Petri interpretată, sigură, viabilă şi fără conflicte structurale. In acest caz se renunţă însă la toate facilităţile pe care le oferă grafcetul. De fapt alegerea ar trebui să se facă în funcţie de complexitatea sistemului modelat. Pentru sisteme relativ simple nu are importanţă modelul folosit. La modelarea sistemelor complexe grafcetul este bun drept caiet de sarcini, dar pentru analiza, sinteza şi validarea modelului este prudent să se folosească reţeaua Petri. In felul acesta greşeli greu detectabile pot fi evitate.

Grafcetul are o serie de caracteristici care îl deosebesc de reţeaua Petri şi îl fac deosebit de util în multe circumstanţe, dintre care se remarcă în special elaborarea de caiete de sarcini pentru sisteme discrete logice complexe şi programarea automatelor programabile logice. Astfel grafcetul permite interpretarea tranziţiilor şi în funcţie de unele evenimente interne (endogene) cum ar fi, de exemplu, apariţia stării active sau inactive[87] a poziţiilor. Drept urmare desenul, care are o reprezentare grafică standardizată internaţional, este mai simplu şi mai inteligibil. Grafcetul permite declanşarea simultană a tranziţiilor fără să existe noţiunea de conflict. Aceasta uşurează întocmirea şi folosirea caietelor de sarcini de către nespecialişti. O poziţie activată şi dezactivată simultan rămâne în grafcet activă[88], spre deosebire de reţeaua Petri. In felul acesta se evită ignorarea acţiunilor corespunzătoare poziţiei şi modificarea algoritmului care a stat la baza întocmirii grafcetului. Ca un preţ pentru facilităţile oferite, grafcetul nu posedă un model matematic atât de elaborat ca reţeaua Petri.

5.10. Sinteza sistemelor discrete logice.

Rezolvarea problemei sintezei sistemelor discrete logice constă în determinarea *relaţiilor logice*. Acestea sunt de două tipuri. Primul tip de relaţii logice determină variabilele de stare prin funcţii logice de timp şi de evenimentele de intrare. Al doilea tip de relaţii logice determină variabilele de ieşire prin funcţii

[87] Reţeaua Petri nu poate testa dacă o poziţie nu este activă.
[88] Prioritate la pornire.

logice de variabilele de stare şi de evenimentele de intrare. Dacă la sistemele logice combinaţionale sinteza se făcea pe baza tabelului de adevăr, la sistemele discrete logice sinteza se face pe baza modelului tip *reţea Petri conformă* sau tip *grafcet*. Deoarece cele două tipuri de modele sunt, după cum s-a arătat, identice în cele mai multe din cazurile întâlnite frecvent în practică, în continuare modelul sistemului discret logic va fi denumit simplu *reţea Petri*.

Există multe metode de sinteză a sistemelor discrete logice. În această lucrare se va pun accent în special pe proiectarea sistemelor discrete logice tip automat finit implementate cu contacte şi relee sau a sistemelor discrete logice tip grafcet implementate cu automate programabile logice.

Prima etapă a sintezei constă în întocmirea caietului de sarcini. În funcţie de complexitatea sistemului caietul de sarcini poate fi o reţea Petri sau un grafcet. În ambele cazuri este recomandabil să se tracă la reţeaua Petri echivalentă care poate fi analizată mai uşor şi permite descoperirea multor erori care nu pot fi evidenţiate pe grafcet.

Urmează, în a doua etapă a sintezei, efectuarea *analizei structurale şi comportamentale* a caietului de sarcini. Dacă acesta este sub formă de grafcet şi urmează să fie implementat pe automate programabile logice, se preferă uneori analiza prin simulare după implementare. Această atitudine se datorează faptului ca grafcetul se poate implementa automat[89]. Totuşi analiza pe reţeaua Petri, inclusiv prin simulare, este mult mai sigură în privinţa evitării unor tipuri de erori.

Reţeaua Petri se *codifică* în etapa următoare a sintezei. Dacă reţeaua este de tip automat finit şi se implementează cu contacte şi relee se foloseşte codul Gray. Dacă reţeaua este de tip grafcet, se implementează pe automat programabil logic şi nu se foloseşte facilitatea de programarea automată se face o codificare distribuită, tip 1 din n.

Relaţiile logice se determină în funcţie de tipul implementării. Dacă se folosesc contacte şi relee se foloseşte diagrama Karnaugh cu variabile incorporate. Dacă implementarea se face pe un automat programabil logic[90] se folosesc un grup de ecuaţii care modelează grafcetul.

5.11. Implementarea sistemelor discrete logice.

După sinteză se face implementarea sistemelor discrete logice. Pentru aceasta se folosesc relaţiile logice stabilite, relaţiile fizice ale componentelor cu care se face implementarea şi relaţiile de echivalenţă între variabilele. Deoarece sinteza depinde, în oarecare măsură, de tipul de implementare ales, cele două probleme vor fi rezolvate împreună. În continuare se vor considera implementarea cu contacte şi relee, cu circuite integrate logice şi cu automat programabil. Implementările cu contacte şi relee sau cu automat programabil sunt cele mai frecvente în domeniul conducerii instalaţiilor pe baza unor modele discrete logice.

---

[89] Implementarea automată este evitată de unii profesionişti. Întocmirea unui program pentru un automat programabil logic nu este aşa de dificilă sau laborioasă şi poate conduce la rezultate mult mai bune decât programarea automată. In plus, dacă sistemul nu funcţionează bine se ştie foarte bine cine este vinovatul!

[90] In cazul în care nu se doreşte sau nu se poate face programarea automată direct pe baza grafcetului.

Implementarea cu circuite integrate logice este prezentată pentru a înţelege mai bine funcţionarea automatului programabil logic şi pentru a aprecia avantajele şi dezavantajele celorlalte două tipuri de implementări .

5.12. Automate elementare cu contacte şi releu.

Cele mai simple sisteme discrete logice au numai două stări şi din această cauză se numesc adeseori *elementare* sau *bistabile*. Reţelele viabile[91] care modelează aceste sisteme sunt în număr de patru şi sunt prezentate în primul rând din *Fig. 5.15*. Alte combinaţii de arce între cele două poziţii (stări) mai sunt posibile, dar reţelele obţinute sau pot fi reduse la cele patru tipuri prezentate sau nu mai sunt viabile şi prin urmare nu pot modela sisteme discrete logice reale. Am codificat denumirea celor două poziţii cu ajutorul variabilei de stare *x*. Poziţia iniţial activă ***A*** are variabila de stare *x=0* iar cea de-a doua poziţie ***B*** este codificată cu variabila de stare *x=1*, după cum se vede în *Fig. 5.15*.

Aplicaţia cea mai răspândită[92] a unora dintre aceste bistabile, în varianta implementării cu contacte şi relee, o constituie automatele pentru pornirea directă a motoarelor electrice. Din această cauză am dat variabilelor care interpretează poziţiile şi tranziţiile denumiri legate de această aplicaţie.

Poziţiile reţelei sunt interpretate de variabila de ieşire *y*. Valoarea *y=0* a variabilei de ieşire corespunde comenzii de motor oprit, iar valoarea *y=1* a variabile de ieşire corespunde comenzii de motor pornit.

Evenimentele care interpretează tranziţiile α, β, γ, δ sunt acţionarea butoanelor de pornire *p* şi respectiv oprire *o*.

[91] Fără blocări.

[92] Alte aplicaţii sunt de asemenea frecvente, de exemplu regulatorul bipoziţional sau blocul de memorare din cadrul unor sisteme discrete logice mai complexe.

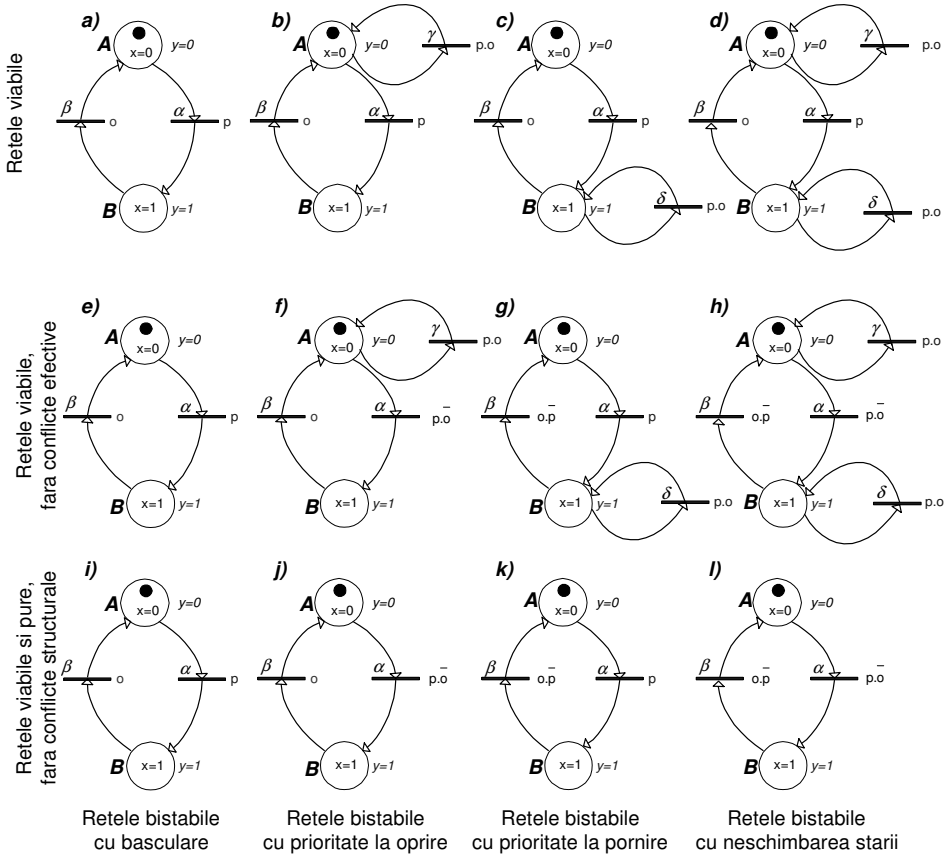


Fig. 5.15 Analiza structurală a celor patru variante posibile de rețele bistabile viabile. Transformarea lor în rețele viabile şi pure, fără conflicte structurale.

5.12.1. Analiza structurală.

Analiza structurala a rețelelor viabile din *Fig. 5.15* b,c,d) indică existența unor conflicte structurale. De exemplu, la rețeaua din *Fig. 5.15* b) din poziția inițială ***A*** pleacă două arce. Unul către tranziția α şi altul către tranziția *γ*. Aceste conflicte devin şi efective atunci când se acționează simultan butoanele de pornire *p* şi oprire *o*. Eliminarea conflictelor efective prin interpretare este prezentată în *Fig. 5.15* f,g,h). De exemplu, conflictul din *Fig. 5.15* b) a fost eliminat interpretând altfel tranziția *α*. În loc ca aceasta să se declanşeze la apăsarea butonului *p* ea se va declanşa, ca în *Fig. 5.15* f), atunci când se apasă butonul *p* şi nu se apasă butonul *o*. În felul acesta, la apăsarea simultană a butoanelor *p* şi *o* se declanşează o singură tranziție γ, cea interpretată de evenimentul *p.o*.

Conflictele efective au dispărut, dar rețelele din *Fig. 5.15* f,g,h) sunt în continuare degenerate pentru că au arce care pleacă şi se întorc din aceiaşi poziție. Se observă însă că prin interpretarea făcută tranzițiilor, pentru eliminarea conflictelor efective, aceste arce nu mai sunt necesare şi pot fi eliminate. Într-adevăr, aceste arce indică faptul că la apăsarea simultană pe butoanele de pornire

şi oprire *p* şi *o*, adică atunci când *p.o=1*, poziţiile din care pleacă şi se întorc arcele rămân active. Dacă eliminăm aceste arce, la apariţia evenimentului *p.o=1* se observă că, datorită interpretării făcute, poziţiile respective rămân active.

5.12.2. Analiza comportamentală.

Toate reţelele viabile şi pure, fără conflicte structurale din ultimul rând al *Fig. 5.15* au următoarea comportare.

În poziţia ***A,*** activă iniţial, variabila de stare este *x=o,* variabila de ieşire este *y=0,* şi motorul este oprit. În această situaţie este validată numai tranziţia *α.*

Dacă evenimentul care constă din apăsarea butonului de pornire apare, adică *p=1*, atunci tranziţia α care este interpretată de acest eveniment se declanşează, poziţia precedentă ***A*** se dezactivează, iar poziţia posterioară ***B*** se activează. În consecinţă *y=1* iar motorul porneşte. În această nouă situaţie în care s-a ajuns este validată numai tranziţia β.

Atunci când se apasă butonul de oprire variabila logică *o=1* şi tranziţia β pe care o interpretează se declanşează deoarece este validată. Iarăşi poziţia precedentă ***B*** se dezactivează iar poziţia posterioară ***A*** se activează şi motorul se opreşte. S-a ajuns în situaţia de la care s-a pornit, deci reţelele analizate realizează performanţa de *reversibilitate.*

În urma analizei comportamentale se observă că reţelele nu se blochează, deci au performanţa de *viabilitate* şi nu au niciodată mai mult de o marcă în fiecare poziţie, deci au performanţa de *siguranţă*.

La apăsarea simultană a butoanelor de pornire şi oprire, adică atunci când *p.o=1* comportarea reţelelor din ultimul rând al *Fig. 5.15* nu mai este aceiaşi.

Pentru reţeaua din *Fig. 5.15* i) la această manevră se produce o basculare. Dacă era activă poziţia **A** cu *x=0* după manevră devine activă poziţia ***B*** cu *x=1* şi invers. Din această cauză reţeaua este denumită bistabilă cu basculare.

Pentru reţeaua din *Fig. 5.15* j) la apăsarea simultană a butoanelor de pornire şi oprire devine activă poziţia ***A*** cu *x=0* şi motorul se opreşte, indiferent în ce situaţia se face această manevră, adică indiferent care poziţie este activă. Se constată deci o prioritate a comenzii de oprire şi din această cauză reţeaua se numeşte bistabilă, cu prioritate la oprire.

Reţeaua din *Fig. 5.15* l) are o comportare asemănătoare cu reţeaua precedentă, numai că la apăsarea simultană a butoanelor de pornire şi oprire *p.o=1* motorul porneşte, indiferent de starea în care se găseşte, adică indiferent de poziţia ***A*** sau ***B*** care este activă. În consecinţă reţeaua este denumită bistabilă cu prioritate la pornire.

În sfârşit, pentru sistemul logic discret modelat de reţeaua din *Fig. 5.15* m) apăsarea simultană pe butoanele de pornire şi oprire *p.o=1* nu provoacă nici o modificare. Dacă motorul era oprit, adică poziţia ***A*** este activă, rămâne oprit şi dacă era pornit, adică poziţia ***B*** este activă, rămâne pornit. Reţeaua se numeşte bistabilă cu neschimbarea stării.

5.12.3. Automatul elementar cu basculare.

Schema bloc a automatului elementar cu basculare este prezentată în Fig. 6.1 b), iar modelul sub forma unei rețele Petri conforme în *Fig. 5.15* i). Din schema bloc se observă că variabilele de intrare sunt *p* şi *o*, iar variabile de ieşire este *y*. Din modul de funcționare stabilit la analiza comportamentală efectuată în paragraful precedent rezultă că ieşirea depinde de intrări prin intermediul variabilei de stare *x*. Pentru codificarea pozițiilor din *Fig. 5.15* i)[93] rezultă următoarea relație dintre ieşire şi stare[94]:

$$y = x \qquad (5.35)$$

Deci ieşirea este identică cu starea. Din rețeaua Petri prezentată în *Fig. 5.15* i) rezultă că starea automatului depinde de cele două intrări. *p* şi *o*, dar şi de starea precedentă a automatului. De exemplu, pentru următoarele intrări $p=1$ şi $o=1$ starea automatului basculează. Dacă înainte de aplicarea intrărilor era $x=0$, după aplicarea intrărilor devine $x=1$. Din (5.35) rezultă că şi ieşirea basculează.

Pentru întocmirea tabelului de adevăr al automatului cu basculare trebuie să se iau în considerare toate valorile intrărilor p^t şi o^t şi a stărilor x^t la momentul *actual de timp t* şi se determină pe baza rețelei din *Fig. 5.15* i) stările $x^{t+\Delta}$ la momentul *viitor de timp* $t+\Delta$. În coloanele din stânga ale Fig. 5.16 sunt prezentate toate valorile posibile ale variabilelor de intrare *p* şi *o* şi ale variabilei de stare *x* la momentul de timp prezent *t*. Pentru completarea tabelului de adevăr, pe baza rețelei *Fig. 5.15* i), se porneşte întotdeauna de la starea actuală[95], dată de valoarea variabilei de stare x^t din coloana a treia a tabelului. Să completăm pe rând liniile tabelului de adevăr din Fig. 5.16.

În prima linie starea actuală este $x^t=0$, adică ne situăm în poziția ***A,*** activă inițial a rețelei din *Fig. 5.15* i) şi deci $y=0$, iar motorul este oprit. Deoarece variabilele de intrare din prima linie a Fig. 5.16 sunt zero, adică $p^t=0$ şi $o^t=0$, înseamnă că nu se apasă nici un buton. Prin urmare nu se declanşează nici o tranziție a rețelei şi, la momentul de timp imediat următor $t+\Delta$ rămâne activă tot poziția inițială ***A*** cu $x^{t+\Delta}=0$ şi motorul oprit. Se trece valoarea în ultima coloană a tabelului.

---

[93] Adică poziției ***A*** îi corespunde valoarea variabilei de stare $x=0$, iar poziției ***B*** îi corespunde valoarea variabilei de stare $x=1$.

[94] Alte relații sunt posibile. De exemplu în cazul în care poziția inițială este codificată cu $x=1$, iar cealaltă poziție este codificată cu $x=0$, relația dintre ieşire şi stare este $y = \overline{x}$

[95] Starea prezentă, starea la momentul de timp *t*.

p^t	o^t	x^t	$x^{t+\Delta}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

Fig. 5.16 Tabelul de adevăr pentru automatul elementar cu basculare.

Linia a doua a Fig. 5.16indică că starea inițial activă a rețelei este poziția ***B*** cu x^t=1 şi motorul este pornit. În această linie variabilele de intrare sunt zero, adică nu apare nici un eveniment şi deci nu se declanşează nici o tranziție a rețelei şi prin urmare rămâne activă tot poziția ***B*** corespunzătoare motorului pornit. Deci la momentul de timp imediat următor $x^{t+\Delta}$=1 şi se completează în consecință ultima coloană a Fig. 5.16.

În linia a treia şi coloana a treia avem $x^t=0$, deci este activă poziția ***A*** şi motorul este oprit. Se apasă numai butonul de oprire, adică $p=0$ şi $o=1$. Apariția evenimentului $o=1$ nu declanşează însă tranziția β pe care o interpretează deoarece aceasta nu este validată. Pentru a fi validată este necesar să fie activă poziția ***B*** cu $x^t=1$. Prin urmare rămâne activă tot poziția ***A*** şi la momentul de timp imediat următor $x^{t+\Delta}=0$ iar motorul rămâne oprit.

Valorile de intrare ale liniei a patra din Fig. 5.16 indică că este activă poziția ***B*** ($x^t=1$), motorul este pornit şi se apasă butonul de oprire $o^t=1$. Din modelul automatului cu basculare din *Fig. 5.15* i) rezultă că se declanşează tranziția validată β pe care o interpretează evenimentul *o*, deci poziția precedentă ***B*** se dezactivează iar poziția posterioară ***A*** se activează. Se trece în coloana a patra a tabelului valoarea $x^{t+\Delta}=0$. *M*otorul se opreşte.

O situație clasică apare în linia cinci a tabelului. Motorul este oprit şi se apasă pe butonul de pornire. Evident, motorul ar trebui să pornească. Dar să vedem dacă aceasta se întâmplă şi atunci când analizăm comportamental rețeaua conformă din *Fig. 5.15* i). Pentru această linie a tabelului este activă poziția ***A***. Tranziția β este validată. Atunci când se apasă butonul de pornire apare evenimentul $p=1$ şi tranziția β se declanşează. Ca urmare a declanşării poziția ***A*** precedentă tranziției de dezactivează iar poziția ***B*** se activează. Deoarece poziția ***B*** este interpretată de variabila *y* cu valoarea 1 motorul porneşte.

Dacă motorul este pornit şi se apasă butonul de pornire, ca în linia şase, motorul trebuie să rămână pornit. Modelul tip rețea Petri conformă din *Fig. 5.15* i) arată tocmai această comportare. În ultima coloană a tabelului de adevăr se trece deci valoarea $x^{t+\Delta}=1$.

Apăsarea simultană a butoanelor de pornire şi oprire, adică cazul în care *p=1* şi *o=1* este analizată în liniile şapte şi opt ale tabelului de adevăr. Rețeaua Petri arată că în acest caz se produce o basculare, care de altfel a şi dat numele automatului elementar. Dacă era activă poziția ***A*** ($x^t=0$), cu motorul oprit, la apariția evenimentelor (apăsarea butoanelor) se declanşează tranziția α şi devine activă poziția ***B,*** caracterizată de valoarea variabilei de stare $x^{t+\Delta}=1$, şi motorul porneşte.

Situația se inversează în linia opt a tabelului. Este activă poziția ***B*** şi prin apăsarea simultană a butoanelor devine activă poziția ***A.*** Se trece $x^{t+\Delta}=0$ în tabel.

$x^{t+\Delta}$

x^t \ $p^t o^t$	00	01	11	10
0	0	0	1	1
1	1	0	0	1

Fig. 5.17 Diagrama Karnaugh pentru automatul elementar cu basculare

Se observă că tabelul de adevăr din Fig. 5.17 şi rețeaua Petri din *Fig. 5.15* sunt echivalente.

Sinteza relațiilor logice pentru automatul elementar cu basculare se face cu ajutorul diagramei Karnaugh din Fig. 5.17obținute din tabelul de adevăr. Relația logică pentru $x^{t+\Delta}$ se obține la fel ca şi în cazul sistemelor logice combinaționale[96]. Cu grupările din Fig. 5.17 se obține relația (5.36).

Implementarea cu contacte şi releu a automatului elementar cu basculare se face considerând relația fizică[97] a releului (5.37) şi relația de echivalență (5.38) între variabila de stare *x* şi variabila de ieşire a releului, contactul său *k*.

$$x^{t+\Delta} = p^t.\overline{x^t} + \overline{o^t}.x^t \qquad (5.36)$$

$$k^{t+\Delta} = K^t \qquad (5.37)$$

$$x^t = k^t \qquad (5.38)$$

Între relațiile (5.36), (5.37) şi (5.38) se elimină variabilele *x* şi Δ. Se obține relația (5.39). În această relație toate variabilele au valoarea de la momentul de

[96] Se observă că sinteza sistemelor discrete logice se reduce la sinteza sistemelor logice combinaționale.

[97] Această relație a fost prezentată în capitolul precedent.

timp actual *t* şi din această cauză se poate renunța la specificarea timpului ca în (5.40).

$$K^t = p^t.\overline{k^t} + \overline{o^t}.k^t \qquad (5.39)$$

$$K = p.\overline{k} + \overline{o}.k \qquad (5.40)$$

Implementarea cu contacte şi releu a relației logice (5.40) este prezentată în Fig. 5.20. Contactul de ieşire *k* al automatului acționează un semnalizator. Această schemă este mai puțin folosită în practică. Se preferă implementarea automatului cu basculare în varianta cu circuite logice integrate.

$\mathbf{p^t}$	$\mathbf{o^t}$	$\mathbf{x^t}$	$\mathbf{x^{t+\Delta}}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**0**
1	1	1	**0**

Fig. 5.18 Tabelul de adevăr pentru automatul elementar cu prioritate la oprire.

$x^{t+\Delta}$

x^t \ $p^t o^t$	**00**	**01**	**11**	**10**
0	**0**	**0**	**0**	**1**
1	**1**	**0**	**0**	**1**

Fig. 5.19 Diagrama Karnaugh pentru automatul elementar cu prioritate la oprire.

Legăturile inverse din Fig. 5.20 leagă releul K cu contactele sale k situate la intrarea sa. Când se apasă butonul de pornire *p* releul K este acționat prin circuitul 10 şi după intervalul de timp Δ îşi acționează contactul său k din circuitul 20 care se închide. Atunci când butonul de pornire *p* nu mai este acționat circuitul

10 se întrerupe dar releul rămâne acţionat prin circuitul 20 deoarece contactul său *k* este închis. Din această cauză contactul *k* mai este denumit *contact de automenţinere*. Deoarece contactul *k* din circuitul 20 are o acţiune care se adună al acţiunea butonului *p,* legătura inversă corespunzătoare se numeşte *pozitivă*.

Circuitul 10 de acţionare a releului este întrerupt nu numai de încetarea acţionării butonului *p* ci şi de acţionarea contactului *k* normal închis al releului. Deoarece acţiunea acestui contact se opune se opune intrării în releu care tinde să în acţioneze, legătura inversă corespunzătoare se numeşte negativă.

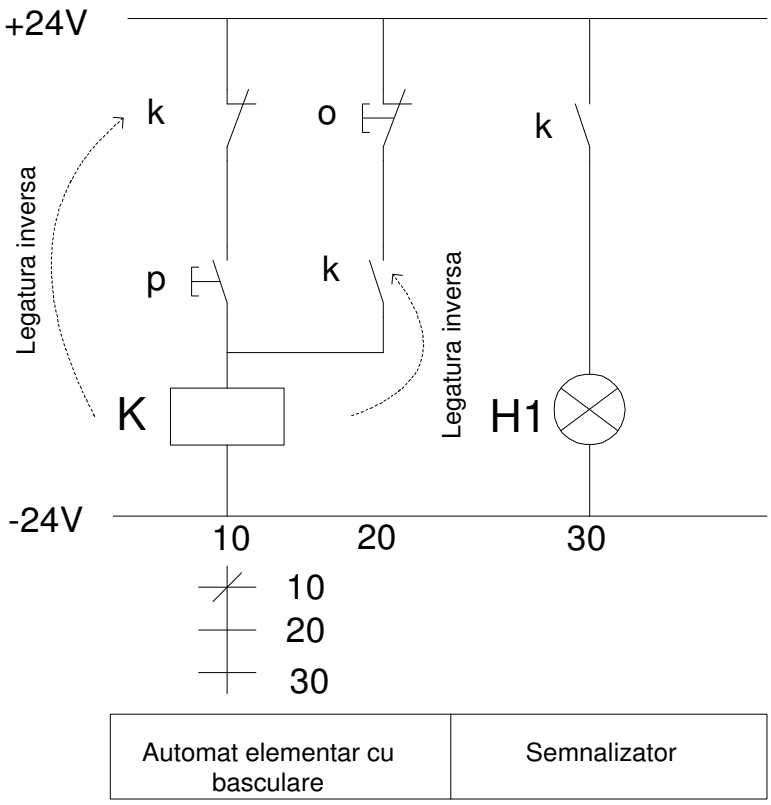


Fig. 5.20 Schema desfăşurată electrică a automatului elementar cu basculare.

În analiza efectuată mai înainte s-a presupus că ambele contacte *k* ale releului din circuitele 10 şi 20 sunt acţionate cu exact aceiaşi întârziere Δ. Dacă întârzierile sunt diferite, ceea ce este cazul în practică, apare fenomenul de *hazard combinaţional*. Rezolvarea diferitor aspecte legate de parametrii fizici ai componentelor sistemului este tratată în capitolul consacrat proiectării sistemelor discrete logice. Sinteza şi implementarea sistemelor discrete logice se referă numai aspectele funcţionale ale sistemului şi presupun caracteristici ideale pentru componentele sale.

5.12.4. Automatul elementar cu prioritate la oprire.

Sinteza şi *implementarea* automatului elementar cu prioritate la oprire se face la fel ca şi în cazul automatului elementar cu basculare din paragraful precedent .

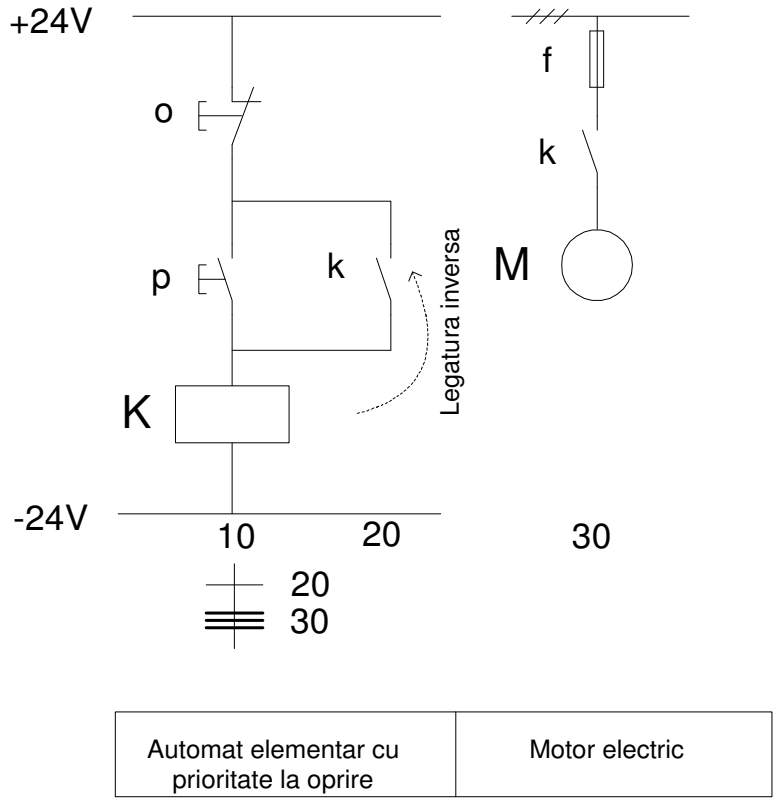


Fig. 5.21 Schema desfăşurată electrică pentru automatul elementar cu prioritate la oprire.

Modelul automatului elementar cu prioritate la oprire este prezentat în *Fig. 5.15* j). Analiza comportamentală a modelului permite completarea tabelului de adevăr din Fig. 5.18. Modificarea care apare în acest caz se referă la liniile şapte şi opt ale tabelului de adevăr. Indiferent care este starea prezentă a reţelei, adică şi în cazul în care este activă poziţia ***A*** cu x^t=0, şi în cazul în care este activă poziţia ***B*** cu $x^t=1$, la apăsarea simultană a butoanelor de pornire ($p^t=1$) şi oprire ($o^t=1$), devine activă poziţia ***A*** cu $x^{t+\Delta}=0$. Această poziţie este interpretată de valoarea variabilei logice de ieşire $y=0$ şi în consecinţă motorul se opreşte. Denumirea automatului provine din faptul că la apăsarea simultană a butoanelor de pornire şi de oprire are *prioritate* butonul de oprire. Această manieră prudentă de tratare a

evenimentelor concurente neprevăzute[98] este adoptată frecvent în practica inginerească.

Diagrama Karnaugh corespunzătoare tabelului de adevăr din Fig. 5.18 este prezentată în Fig. 5.19. Pentru cele două grupări din diagrama Karnaugh se obţine relaţia logică (5.41).

$$x^{t+\Delta} = \overline{o^t}.p^t + \overline{o^t}.x^t = \overline{o^t}.(p^t + x^t) \qquad (5.41)$$

Pentru implementarea cu contacte şi releu se procedează la fel ca în cazul automatului cu basculare prezentat în paragraful precedent. Se scriu relaţia fizică a releului şi relaţia de echivalare a variabilelor li se elimină variabilele *x, Δ* şi *t*. Se obţine relaţia (5.42) care se implementează ca în Fig. 5.21. Contactul de ieşire *k* al automatului acţionează în acest caz[99] un motor electric. Aceasta este o schemă desfăşurată electrică *simplificată*[100] care ilustrează o situaţie frecvent întâlnită în practică. Acest automat este folosit adeseori şi ca o componentă a unor sisteme mai complexe de conducere.

Automatul cu prioritate la oprire, la fel ca şi automatul cu basculare, prezintă o *legătură inversă*[101] *pozitivă* în circuitul 20. Contactul de automenţinere *k* din acest circuit ajută acţiunea butonului *p,* care constă în acţionarea releului K, şi o menţine şi după ce acţiunea acestuia a încetat. Ne reamintim definiţia sistemului automat: un sistem cu memorie sau legătură inversă. În cazul automatelor elementare legătura inversă realizează tocmai acţiunea de memorare[102]. Având în vedere această proprietate cu totul remarcabilă a legăturii inverse se poate afirma că pentru a construi un sistem automat este suficient să introducem legături inverse.

$$K = \overline{o}.(p + k) \qquad (5.42)$$

---

[98] Apăsarea simultană a butoanelor de pornire şi oprire este o manevră care nu este prevăzută de obicei mla funcţionarea normală a instalaţiilor. Din această cauză este mai prudent ca la această manevră instalaţia să se oprească.

[99] Automatul cu prioritate la oprire poate acţiona şi alte echipamente, cum ar fi , de exemplu, o instalaţie electrică de încălzire.

[100] Unele elemente importante ale schemei clasice de pornire directa a unui motor, cum ar fi releul termic de protecţie a motorului, au fost omise, accentul punându-se pe sinteza şi implementarea automatului cu prioritate la oprire. Se observă alimentarea separată a părţii de comandă automată şi a părţii de forţă a instalaţiei.

[101] Feedback (eng.)

[102] Memorarea se poate realiza şi prin alte metode, diferite de metoda legăturii inverse, de exemplu folosind un sistem de came. Şi în prezent programul de acţiuni secvenţiale la majoritatea maşinilor automate de spălat este memorat şi se realizează cu ajutorul unui sistem de came.

p^t	o^t	x^t	$x^{t+\Delta}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

Fig. 5.22 Tabelul de adevăr pentru automatul cu prioritate la pornire.

5.12.5. Automatul elementar cu prioritate la pornire.

Spre deosebire de automatul din paragraful precedent, care se opreşte la apăsarea simultană a butoanelor de pornire şi oprire, acest automat *porneşte*. Se spune că are prioritate butonul de pornire. Tabelul de adevăr întocmit, la fel ca şi la , pe baza modelului sub formă de rețea Petri conformă (*Fig. 5.15* k) este prezentat în Fig. 5.22.

$x^{t+\Delta}$

x^t \ $p^t o^t$	**00**	**01**	**11**	**10**
0	**0**	**0**	**1**	**1**
1	**1**	**0**	**1**	**1**

Fig. 5.23 Diagrama Karnaugh pentru automatul elementar cu prioritate la pornire.

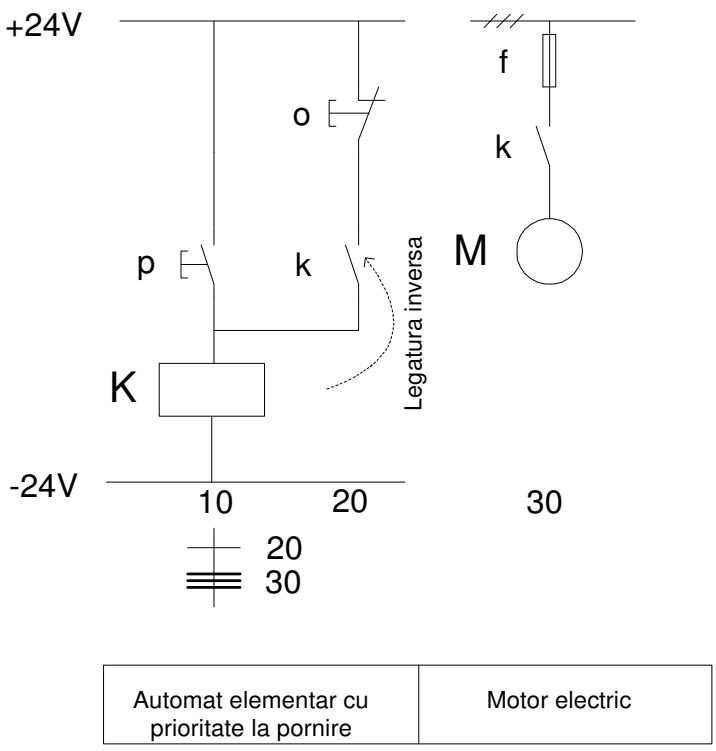


Fig. 5.24 Schema desfăşurată electrică pentru automatul elementar cu prioritate la pornire.

Alegerea priorităţii la pornire se face în liniile şapte şi opt ale tabelului. Dacă $x^t=0$, adică este activă poziţia ***A*** şi motorul este oprit, este validată tranziţia α reţelei din *Fig. 5.15* k). La apariţia evenimentelor simultane $p=1$ şi $o=1$ tranziţia α se declanşează pentru că este validată şi interpretată de evenimentul *p* care a apărut. Poziţia ***A***, precedentă tranziţiei α, se dezactivează şi poziţia ***B***, posterioară tranziţiei α, se activează. Dacă examinăm reţeaua la momentul următor declanşării tranziţiei, adică la timpul $t+\Delta$, constatăm că singura poziţie activă este ***B***. Variabila de starea a automatului ia acum valoarea $x^{t+\Delta}=1$. Deoarece în cazul reţelelor din *Fig. 5.15* variabilele de stare sunt egale cu variabilele de ieşire avem şi $y^{t+\Delta}=x^{t+\Delta}=1$, adică motorul porneşte.

În ultima linie, a opta, a tabelului de adevăr variabila de stare s automatului cu are valoarea $x^t=1$, adică este activă poziţia ***B*** a reţelei din *Fig. 5.15* k) şi motorul este pornit. Deoarece ***B*** este activă, tranziţia β este validată. Pentru reţeaua analizată tranziţia β este interpretată de un eveniment care apare atunci când se apasă butonul de oprire dar nu se apasă butonul de pornire. În linia opt a tabelului de adevăr valorile variabilelor p şi o arată ca în această situaţie ambele butoane sunt apăsate. Deci nu este îndeplinită condiţia de declanşare a tranziţiei β şi poziţia ***B*** rămâne în continuare activă, cu motorul pornit.

$\mathbf{p^t}$	$\mathbf{o^t}$	$\mathbf{x^t}$	$\mathbf{x^{t+\Delta}}$
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**0**
1	1	1	**1**

Fig. 5.25 Tabelul de adevăr pentru automatul elementar cu neschimbarea stării.

Pentru sinteza automatului elementar cu prioritate la pornire datele din tabelul de adevăr Fig. 5.22 se trec în diagrama Karnaugh din Fig. 5.23. Se pot face două grupări şi rezultă următoarea relaţie logică.

$$x^t + \Delta = p^t + \overline{o^t}.x^t \qquad (5.43)$$

După considerarea relaţiei fizice (5.37) a releului şi a relaţiei de echivalenţă a variabilelor (5.38) rezultă relaţia (5.44) care poate fi implementată ca în Fig. 5.24.

$$K = p + \overline{o}.k \qquad (5.44)$$

Automatul cu prioritate la pornire, în varianta implementată cu contacte şi releu, este foarte puţin folosit în practică. În schimb acest automat este de bază la implementarea software pe baza caietului de sarcini grafcet.

Şi automatul cu prioritate la oprire îşi realizează acţiunea de memorare a comenzii de pornire a motorului dată prin butonul *p* cu ajutorul unei *legături inverse* realizate prin intermediul contactului de automenţinere *k* din circuitul 20. Comparând schemele de la automatele cu prioritate la oprire şi cu prioritate la pornire se constată că în primul caz butonul de oprire întrerupe atât acţiunea butonului p cât şi acţiunea contactului de automenţinere *k*, pe când în cel de al doilea caz oprirea întrerupe doar circuitul 20 de automenţinere.

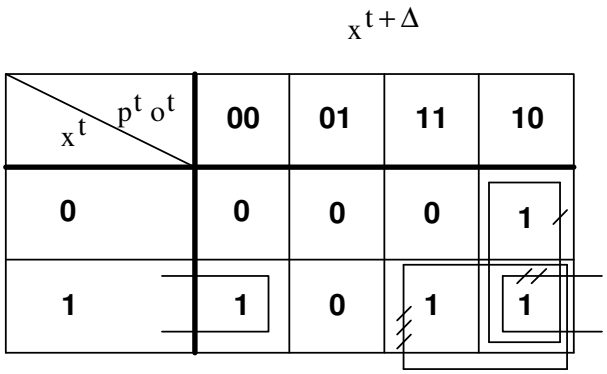


Fig. 5.26 Diagrama Karnaugh pentru automatul elementar cu neschimbarea stării.

5.12.6. Automatul elementar cu neschimbarea stării.

Acest automat nu şi-a găsit încă aplicarea în varianta implementată cu contacte şi releu. Se prezintă totuşi, ca un exerciţiu, sinteza şi implementarea sa.

Paşii urmaţi sunt aceiaşi ca la celelalte automate elementare. Tabelul de adevăr din Fig. 5.25 este obţinut prin analiza comportamentală a reţelei din *Fig. 5.15* l). Se observă că pentru această reţea tranziţiile α şi β sunt astfel interpretate încât nu se declanşează decât atunci când se apasă un buton şi *nu* se apasă celălalt. În consecinţă, la apăsarea simultană a butoanelor de pornire şi oprire tranziţiile nu vor putea fi declanşate chiar dacă sunt validate. Deci la această manevră, apăsarea simultană a butoanelor, reţeaua şi automatul pe care îl modelează îşi păstrează starea. De aici provine şi denumirea automatului elementar. În tabelul de adevăr, pentru coloanele şapte şi opt avem $x^{t+\Delta}=x^t$.

Sinteza automatului se face trecând datele din tabelul de adevăr Fig. 5.25 în diagrama Karnaugh din Fig. 5.26 şi făcând grupări. Relaţia logică corespunzătoare celor trei grupări este dată de (5.45).

$$x^{t+\Delta} = p^t.\overline{o^t} + \overline{o^t}.x^t + p^t.x^t = \overline{o^t}.(p^t + x^t) + p^t.x^t \qquad (5.45)$$

Se adaugă relaţia fizică (5.37) şi relaţia de echivalenţă (5.38) fiind posibilă eliminarea variabilelor *x, Δ* şi *t.* Se obţine relaţia logică finala (5.46)

$$K = \overline{o}.(p+k) + p.k \qquad (5.46)$$

Schema desfăşurată electrică corespunzătoare relaţiei obţinute prin sinteză este prezentată în Fig. 5.27.

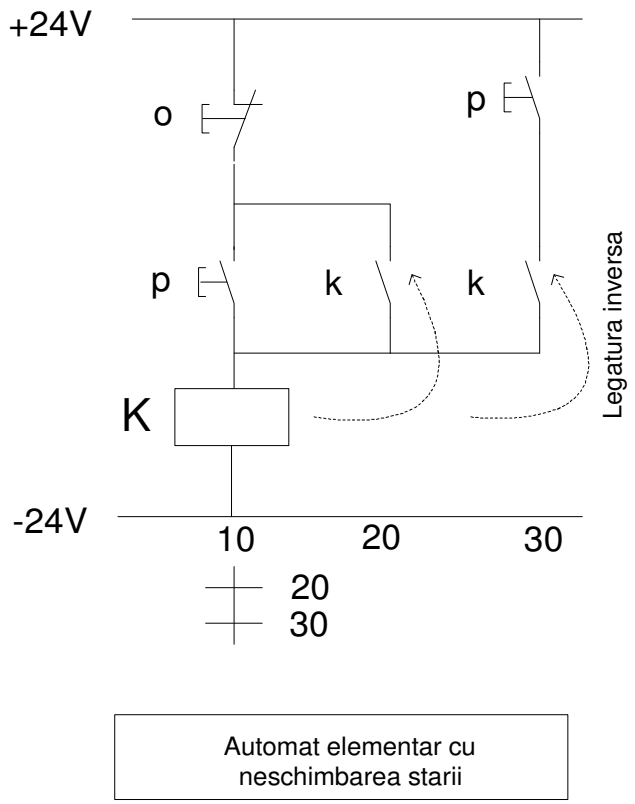


Fig. 5.27 Schema desfăşurată electrică pentru automatul elementar cu neschimbarea stării.

La automatul cu neschimbarea stării există de asemenea două *legături inverse* ca şi la automatul cu basculare. La apăsarea simultană a butoanelor p şi o automatul ar trebui să nu-şi schimbe starea. Acest lucru se şi întâmplă. Dar dacă acțiunea celor două butoane nu este perfect simultană pot apare erori.

5.12.7. Automatul elementar pentru reglarea bipoziționalǎ.

Automatele elementare implementate cu contacte şi releu sunt aplicate frecvent la conducerea sistemelor. Un exemplu îl reprezintă automatul pentru reglarea bipozițională a nivelului într-un rezervor.

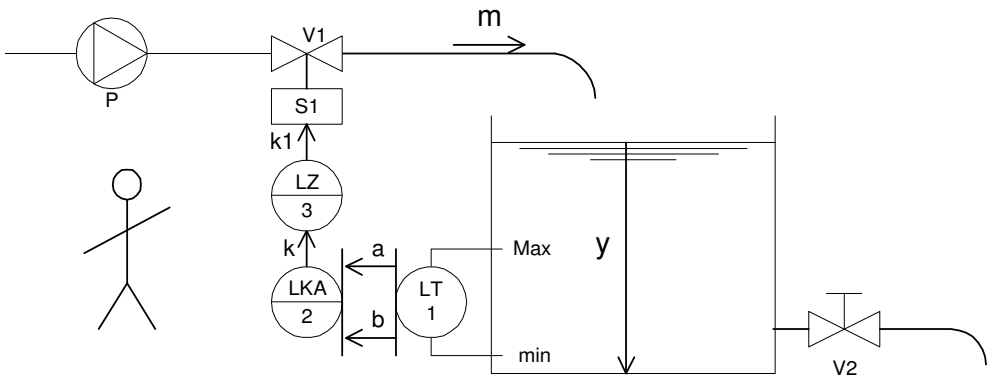


Fig. 5.28 Schema tehnologică cu aparatura de automatizare a sistemului cu automat cu contacte şi relee pentru reglarea nivelului

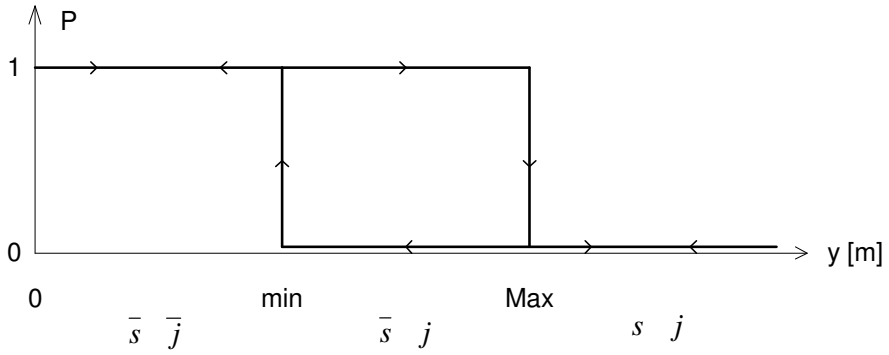


Fig. 5.29 Caracteristica de funcţionare a pompei instalaţiei din Fig. 5.28.

Schema tehnologică a sistemului automat este prezentată în Fig. 5.28. În acest paragraf se prezintă sinteza şi implementarea cu contacte şi releu a automatului pe baza modelului sub forma unei reţele Petri conforme.

Reglarea automată a nivelului în rezervor se face, în exemplul considerat, pe baza unui algoritm bipoziţional care porneşte sau opreşte pompa în funcţie de nivelul măsurat în rezervor. Traductoarele sistemului au la ieşire un contact *s* care este acţionat *(s=1)* atunci când nivelul *y* depăşeşte valoarea **Max** şi un contact *j* care este acţionat *(j=1)* atunci când nivelul *y* depăşeşte valoarea **min.** Caracteristica de funcţionare a pompei, conform acestui algoritm, este prezentată în Fig. 5.29.

În graficul din Fig. 5.29 se disting trei zone. Pentru zona în care nivelul $y \geq Max$ contactele *s* şi *j* sunt acţionate[103] *(s=1, j=1)*, iar pompa este oprită *(P=0).* În zona în care nivelul $y < min$ contactele *s* şi *j* **nu**[104] sunt acţionate *(s=0, j=0),* iar pompa este pornită *(P=1).* Zona intermediară caracterizată de nivelul $min \leq y < Max$ este sesizată de traductoare astfel: contactul a nu este acţionat *(s=0)* şi contactul b este acţionat *(j=1).* În *această zonă* pompa este pornită sau oprită în funcţie de starea sa precedentă. Dacă era pornită rămâne pornită şi dacă era oprită rămâne oprită. Caietul de sarcini al automatului care porneşte şi opreşte pompa conform acestui algoritm este prezentat sub forma unei reţele Petri în Fig. 5.30 a).

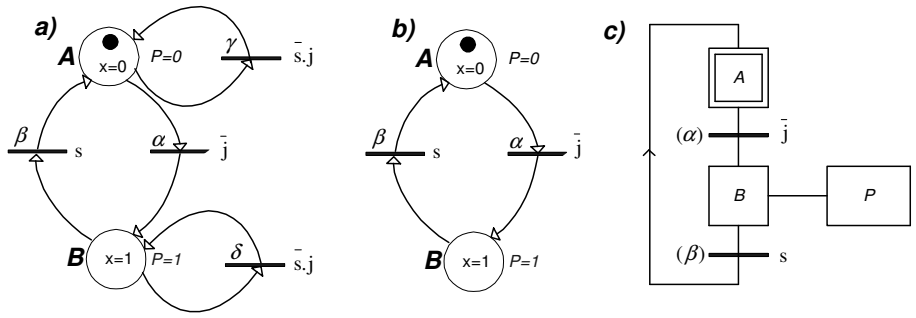


Fig. 5.30. Variante ale caietului de sarcini pentru automatul secvenţial care reglează bipoziţional nivelul într-un rezervor.

Se observă că automatul are structura tipică a unui automat cu neschimbarea stării. *Analiza structurala* a reţelei arată că conflictele structurale care apar în poziţiile ***A*** şi ***B*** nu sunt şi efective datorită evenimentelor care interpretează tranziţiile *γ* şi *δ.* Din această cauză buclele care pornesc din poziţiile ***A*** şi ***B*** pot fi înlăturate rezultând reţeaua Petri din Fig. 5.30 b). Această reţea fiind *conformă* are grafcetul echivalent din Fig. 5.30 c).

Sinteza automatului se face pe baza reţelei din Fig. 5.31 b). La început se *codifică* poziţiile. Având numai două poziţii variabila de stare poate fi codificată cu ajutorul unui singur bit *x*. Deoarece în A pompa este oprită se ia valoarea *x=0*. Din acelaşi motiv se atribuie lui ***B*** valoarea *x=1. Analizând comportamental* reţeaua se poate completa tabelul de adevăr din Fig. 5.31. În primele două linii ale tabelului evenimentele s şi j nu au apărut[105]. Această situaţie coincide zonei din Fig. 5.29 în care nivelul este mai mic decât valoarea minimă. Dacă poziţia A a automatului este activă este validată tranziţia α şi deoarece evenimentul *j* nu a apărut avem relaţia (5.47).

$$\overline{j} = 1 \qquad (5.47)$$

[103] Evenimentele *a* şi *b* au apărut, în exprimarea specifică reţelelor Petri conforme.
[104] În Fig. 5.29 evenimentele a şi b sunt negate, indicând faptul că nu au apărut.
[105] Variabilele logice corespunzătoare evenimentelor au valorile s=*0* şi *j=0.*

Expresia logică care interpretează tranziția α are valoarea 1 şi tranziția se declanşează deoarece este validată. Poziția ***A*** devine inactivă iar poziția ***B*** devine activă. Deci $x^{t+\Delta}=1$ şi se completează în mod corespunzător ultima coloană a tabelului de adevăr. În mod asemănător se completează toată coloana a patra a tabelului.

s^t	j^t	x^t	$x^{t+\Delta}$
0	0	0	**1**
0	0	1	**1**
0	1	0	**0**
0	1	1	**1**
1	0	0	**-**
1	0	1	**-**
1	1	0	**0**
1	1	1	**0**

Fig. 5.31 Tabelul de adevăr al automatului pentru reglarea bipoziţională a nivelului în rezervor.

O problemă apare în liniile cinci şi şase ale tabelului de adevăr. Varianta în care *s=1* şi *j=0* nu este posibilă din punct de vedere fizic. Nivelul nu poate fi în acelaşi timp mai mare decât valoarea maximă şi mai mic decât valoarea minimă. Din acest motiv, pentru aceste linii în coloana a patra a tabelului de adevăr nu s-a trecut nici o valoare a variabilei de stare $x^{t+\Delta}$.

$x^{t+\Delta}$

x^t \ $s^t j^t$	00	01	11	10
0	1	0	0	-
1	1	1	0	-

Fig. 5.32 Diagrama Karnaugh a automatului pentru reglarea bipoziţională a nivelului.

Valorile din tabelul de adevăr Fig. 5.31 se trec în diagrama Karnaugh Fig. 5.32 şi se fac două grupări. Expresia logică rezultantă este dată de

$$x^{t+\Delta} = \overline{j^t} + \overline{s^t}.x^t$$

Considerând relaţia fizică a releului (5.37) şi relaţia de echivalenţă (5.38) se pot elimina variabilele *x, Δ* şi *t* rezultând relaţia (5.48).

$$K = \bar{j} + \bar{s}.k \qquad (5.48)$$

Implementarea cu contacte şi relee a aceste relaţii este prezentată în schema desfăşurată electrică din Fig. 5.33, circuitele 10 şi 20. Contactul ***a*** corespunde variabilei ***s*** ce caracterizează nivelul ***Max*** iar contactul ***b*** corespunde variabilei ***j*** ce caracterizează nivelul ***min***.

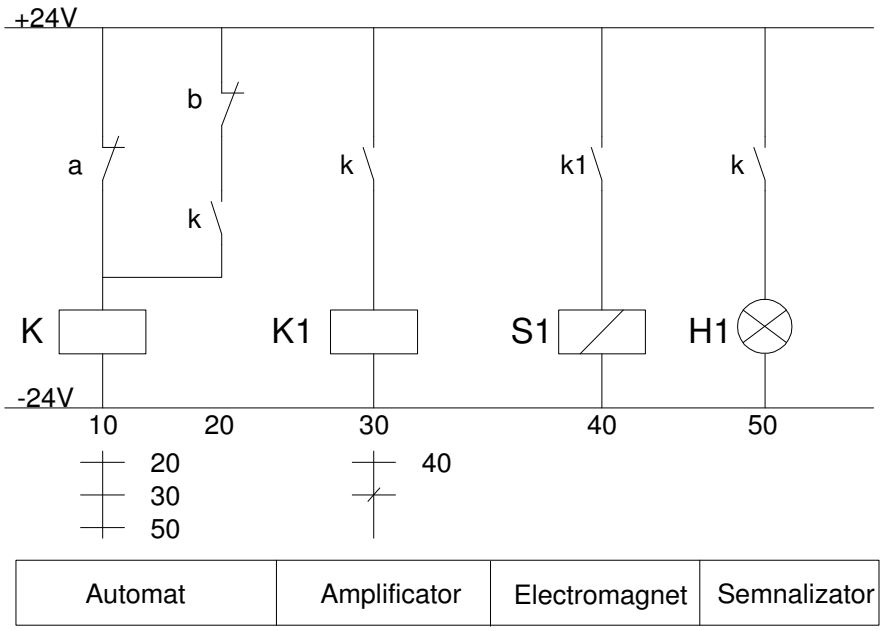


Fig. 5.33 Schema desfăşurată electrică simplificată pentru automatul 2, elementul de execuţie 3 şi electromagnetul S1 din Fig. 5.28

5.13. Automate elementare cu circuite integrate

Ca şi sistemele logice combinaţionale, sistemele discrete logice simple, bistabile, pot fi implementate software sau cu circuite integrate logice. Sistemele bistabile implementate cu circuite electronice integrate poartă numele de *circuite bistabile* şi au proprietăţi cu totul remarcabile care au condus la folosirea lor intensă nu numai în domeniul sistemelor de conducere dar şi la sistemele de comunicaţie sau la sistemele de calcul. Bistabilele pot fi asincrone sau sincrone.

5.13.1. Bistabilul RS.

În paragraful 5.12.5 am sintetizat relația logică (5.43) corespunzătoare rețelei bistabile cu prioritate la pornire din *Fig. 5.15* k). Pentru implementarea sa cu circuite integrate logice, la fel ca şi în cazul implementării cu contacte şi releu, trebuiesc adăugate relația fizică a circuitului integrat şi relația de echivalare a variabilelor.

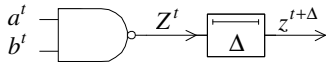


Fig. 5.34 Schema bloc a unui circuit integrat real.

Semnalul printr-un circuit electronic real nu se propagă instantaneu de la intrare la ieşire. Există o mică întârziere Δ care poate fi luată în considerare cu ajutorul unui bloc nou conectat la ieşirea circuitului ca în schema bloc din Fig. 5.34. Această caracteristică poate fi pusă în evidență, la fel ca şi în cazul releului, cu ajutorul relației (5.49).

$$z^t + \Delta = Z^t \tag{5.49}$$

Legătura dintre variabilele modelului sub formă de rețea Petri şi variabilele circuitului integrat este dată de relația (5.50). Eliminând variabilele *x, Δ* şi *t* din relațiile (5.43), (5.49) şi (5.50) rezultă (5.51).

$$x^t = z^t \tag{5.50}$$

$$Z = p + \bar{o}.z \tag{5.51}$$

Pentru a putea fi implementată cu ajutorul circuitelor ŞI – NU o relație scrisă în forma disjunctivă canonică trebuie[106] negată, trebuie să i se alice teorema De Morgan şi apoi trebuie negată din nou. Aplicând aceste operații relației (5.51) se obține (5.52) care poate fi implementată ca în *Fig. 5.35*.

$$Z = \overline{\bar{p}.\overline{(\bar{o}.z)}} \tag{5.52}$$

[106] La fel ca şi în cazul sistemelor logice combinaționale.

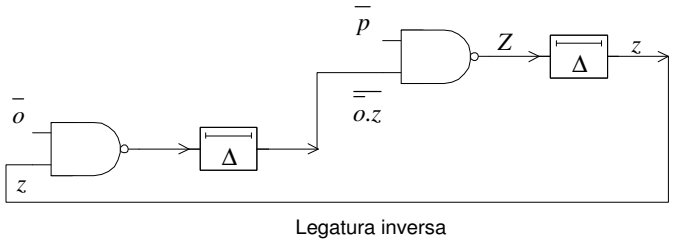


Fig. 5.35 Implementarea automatului cu prioritate la pornire cu circuite integrate ŞI – NU reale.

În *Fig. 5.35* se fac următoarele simplificări:

- Toate blocurile de întârziere sunt aduse la ieşirea circuitului formând un singur bloc care nu mai este reprezentat în desen. Această simplificare face ca sistemul obţinut să fie diferit de cel din *Fig. 5.35*. Aici cele două intrări au timpi de întârziere diferiţi pe drumul lor de la intrarea la ieşirea circuitului. După simplificare întârzierile sunt acelea şi pentru cele două semnale de intrare.
- Se redesenează schema, aducând circuitul ŞI – NU de jos în dreptul celui de sus Fig. 5.36.
- Se consideră că apăsarea simultană pe butoanele de pornire şi de oprire este imposibilă.
- Semnalul de intrare, butonul de pornire *p,* este redenumit *S,* de la englezescul *Set.* Asemănător semnalul *o* devine *R,* de la *Reset*. Semnalul de ieşire *z* este redenumit *Q*.
- Se specifică pe desen o a doua ieşire, de la circuitul ŞI – NU de jos. În condiţiile în care nu este posibil evenimentul unei apăsări simultane pe butonul *p* şi pe butonul *o*, cea de a doua ieşire este egală cu negata primei ieşiri.
- Tot în absenţa apariţiei simultane a celor două semnale de intrare se observă că *R* este egal cu negatul lui *S* şi invers.

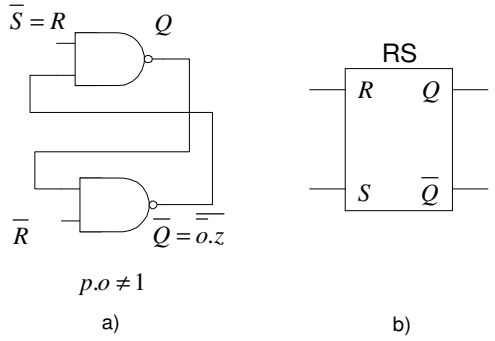


Fig. 5.36 Circuitul bistabil RS şi simbolul său.

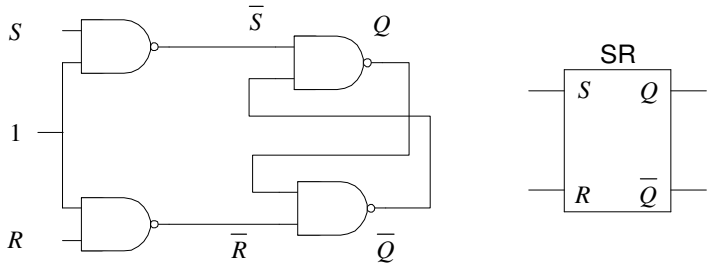


Fig. 5.37 Bistabilul SR şi simbolul său.

Rezultatul simplificărilor apare în Fig. 5.36.a). Cele doua circuite ŞI – NU sunt implementate într-un singur modul cu simbolul din Fig. 5.36 b). Circuitul bistabil are aceleaşi funcţiuni ca şi automatele elementare, memorează impulsurile de pornire (setare) sau oprire (resetare).

5.13.2. Bistabilele SR şi SRC. Sincronizarea.

Neajunsul că intrările bistabilului RS sunt negate se poate înlătura uşor cu ajutorul a două circuite ŞI – NU ca în Fig. 5.37 şi se obţine bistabilul SR.

Circuitele electronice sunt foarte rapide şi răspund la cele mai rapide perturbaţii. Comparativ cu acestea releul are o inerţie considerabilă care îl face imun la perturbaţiile de înaltă frecvenţă. Acelaşi efect se obţine la bistabile introducând *sincronizarea*.

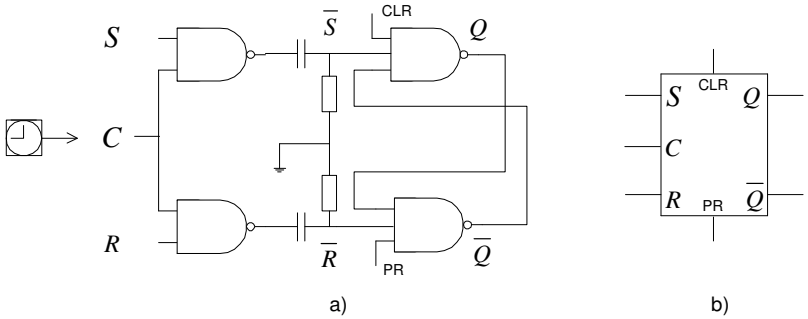


Fig. 5.38 Bistabilul SRC şi simbolul său.

În Fig. 5.38 a) s-a introdus o bornă suplimentară, de ceas[107], la care se trimit impulsuri periodice. Datorită condesatoarelor introduse în schemă comenzile de setare *S* şi resetare *R* sunt luate în considerare numai pe frontul crescător al impulsurilor de ceas. În felul acesta orice impuls parazit care apare între fronturile crescătoare ale impulsurilor de ceas sunt ignorate. Bornele CLEAR şi PRESET au

[107] Clock (eng.)

acelaşi efect cu S negat şi, respectiv, R negat. Semnalele aplicate la aceste borne aduc ieşirea Q la nivelul 0, respectiv 1, într-un mod asemănător bistabilului RS.

5.13.3. Bistabilul SCR Master – Slave.

O măsură suplimentară de protecţie la impulsuri perturbatoare care poate fi luată în cazul circuitelor integrate constă în ruperea legăturii electrice directe dintre intrare şi ieşire. Pentru aceasta se folosesc două bistabile SRC ca în Fig. 5.39. Primul bistabil funcţionează pe frontul crescător al impulsului de ceas şi al doilea bistabil transmite semnalul la ieşire pe frontul descrescător al impulsului de ceas.

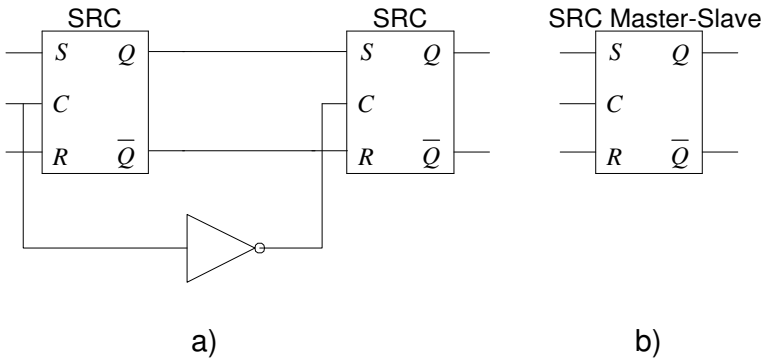


Fig. 5.39 Bistabilul SRC Master – Slave şi simbolul său.

5.13.4. Bistabilul JKC.

Dacă se introduc două legături inverse suplimentare la un bistabil SRC, aşa cum se vede în Fig. 5.40, se obţine bistabilul JKC.

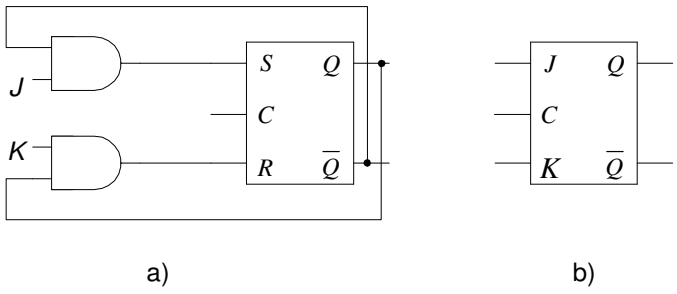


Fig. 5.40 Bistabilul JKC şi simbolul său.

Borna la care se aplică semnalul de oprire se numeşte acum J[108] iar borna la care se aplică semnalul de oprire se numeşte K[109]. Datorită legăturilor inverse trecerea circuitului dintr-o stare în alta, bascularea sa , se face acum mult mai rapid decât în cazul bistabilului SRC. Mai mult, la aplicarea simultană a semnalului de pornire şi oprire, $J=1$ şi $K=1$, circuitul basculează, la fel ca şi în cazul automatului elementar cu contacte şi releu care are drept model reţeaua Petri din *Fig. 5.15*. Relaţia logică (5.36) se transformă cu noile notaţii în (5.53)

$$Q^{t+\Delta} = J^t.\overline{Q^t} + \overline{K^t}.Q^t \qquad (5.53)$$

5.13.5. Bistabilul D.

O conexiune ca în Fig. 5.41 transformă bistabilul JKC într-un bistabil cu o singură intrare *D*. Dacă se înlocuieşte în (5.53) *J* cu *D,* iar *K* cu negatul lui *D* se obţine relaţia logică (5.54). Această relaţie este identică cu relaţia fizică a releului, deci bistabilul D poate realiza aceleaşi funcţiuni logice ca un releu.

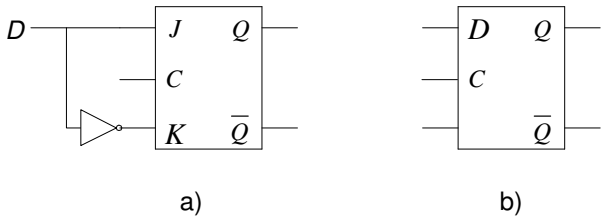


Fig. 5.41 Bistabilul D şi simbolul său.

$$Q^{t+\Delta} = D^t.\overline{Q^t} + \overline{\overline{D^t}}.Q^t = D^t \qquad (5.54)$$

5.13.6. Bistabilul T.

O legătură simpla ca în Fig. 5.42 transformă bistabilul JKC în bistabilul T.

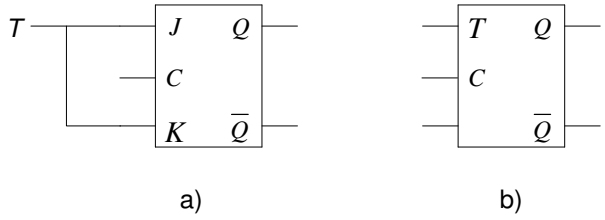


Fig. 5.42 Bistabilul T şi simbolul său.

[108] Jam (eng.)
[109] Keep (eng.)

Înlocuind în (5.54) noile valori ale lui J şi K se obţine relaţia logică (5.55) a bistabilului T. Dacă *T=0* ieşirea bistabilului rămâne neschimbată. Dacă *T=1* ieşirea basculează, adică dacă era 0 devine 1 şi dacă era 1 devine 0.

$$Q^{t+\Delta} = T^t.\overline{Q^t} + \overline{T^t}.Q^t = T^t \oplus Q^t \qquad (5.55)$$

5.14. Sinteza automatelor implementate cu contacte şi relee

Metoda de sinteză va fi prezentată pe baza exemplului automatului care porneşte un motor electric şi după o rotaţie să îl opreşte în poziţia iniţială.

5.14.1. Schema tehnologică şi schema bloc

Primul pas în proiectarea unui sistem automat constă în alegerea soluţiei de conducere automată[110]. În cazul de faţă se doreşte o soluţie simplă, care să nu controleze continuu poziţia axului motorului. Motorul trebuie să pornească la acţionarea butonului de pornire *p* şi să se oprească la acţionarea contactului *a* care indică terminarea unei rotaţii. Algoritmul de conducere prezentat ia în considerare numai evenimente discrete logice şi prin urmare sistemul automat va fi de tip *discret logic*.

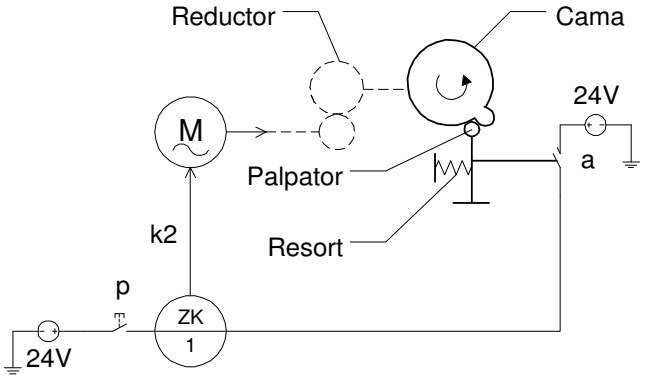


Fig. 5.43Schema tehnologica cu aparatura de automatizare a sistemului automat discret logic pentru oprirea motorului electric după o rotaţie.

Automatistul împreună cu tehnologul[111] elaborează schema tehnologică cu aparatura de automatizare din Fig. 5.43 şi schema bloc din Fig. 5.44. Modul în care se realizează aceasta nu poate fi algoritmizat. Soluţia aleasă depinde de foarte mulţi factori: banii disponibili pentru instalaţia de automatizare, aparatele de automatizare care pot fi procurate, mediul de lucru al sistemului, fiabilitatea impusă sistemului, calificarea necesară personalului de întreţinere, costul şi timpul

[110] Alegerea soluţiei de automatizare.
[111] Adeseori sunt una şi aceeaşi persoană.

acordat proiectului de automatizare, etc. In cazuri simple, ca acesta, se folosesc exemple asemănătoare[112].

[112] Metoda de proiectare bazată pe exemple tipice.

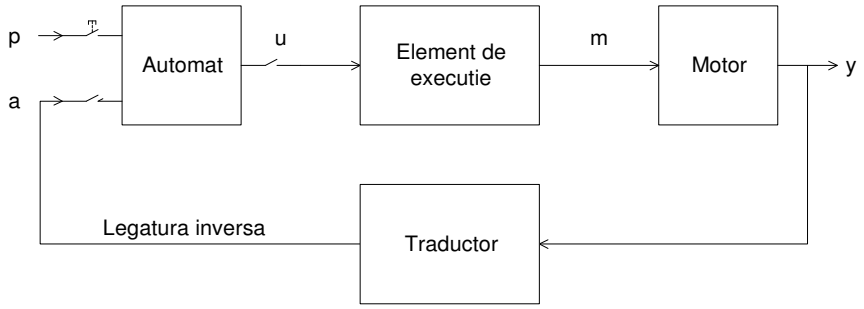


Fig. 5.44 Schema bloc a sistemului automat pentru oprirea motorului electric după o rotaţie.

Schema tehnologică cu aparatura de automatizare din Fig. 5.43 prezintă instalaţia care urmează să fie condusă automat, motorul electric împreună cu traductorul de poziţie şi staţia de comandă cu numărul1.

Traductorul de poziţie este format din reductorul de turaţie, cama, palpatorul şi contactul *a.* La terminarea unei rotaţii palpatorul acţionează contactul *a* care se închide. Motorul se mai roteşte puţin şi datorită formei camei palpatorul deschide contactul *a*. În această poziţie motorul trebuie să se oprească. Traductorul trebuie ales după caracteristicile tehnice necesare din cataloagele de specialitate sau trebuie proiectat şi construit.

Staţia de comandă a poziţiei motorului are la intrare contactele *a* şi *p,* şi la ieşire contactul *k2*. Simbolul care o reprezintă indică faptul că staţia de comandă se găseşte în alt loc decât motorul electric. Distanţa dintre staţie şi motor poate ajunge la câteva sute de metri. Din punct de vedere fizic staţia de comandă este un dulap, cutie sau tablou care conţine diferite componente şi aparate de automatizare. In mod curent staţia de comandă poate conţine:

- Alimentarea cu energie electrică a instalaţiei tehnologice.
- Alimentarea instalaţiei de automatizare.
- Automatul, automatul programabil, regulatoarele sau microcalculatorul de proces.
- Sistemul de semnalizare.
- Sistemul de protecţie.
- Sistemul de comutare în diferite regimuri de funcţionare.
- Sistemul de comunicare la distanţă.

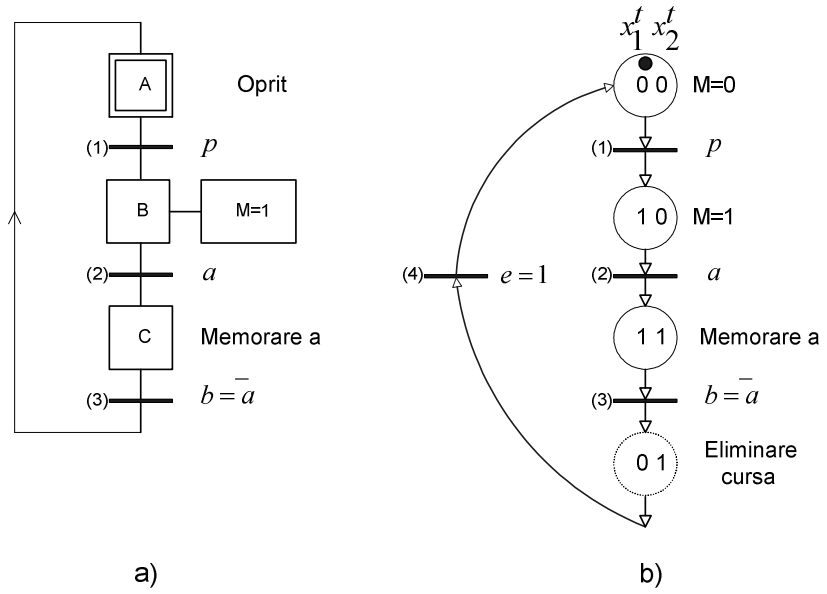


Fig. 5.45 Grafcetul şi rețeaua Petri echivalentă a automatului pentru oprirea unui motor după o rotație.

Stațiile de comandă trebuiesc proiectate şi construite din punct de vedere hardware (mecanic, electric şi electronic) şi software. În această lucrare se va pune accentul, în special, pe proiectarea sistemului de conducere.

Schema bloc a sistemului automat din Fig. 5.44 pune în evidență componentele principale ale sistemului de conducere automată. Comparând schema bloc cu schema tehnologică din Fig. 5.43 se constată că stația de comandă conține, în acest caz automatul şi elementul de execuție. Motoarele electrice de putere mare nu pot fi pornite oricum. Există un procedeu de pornire care poate fi implementat cu contacte şi relee[113] sau cu ajutorul unui aparat electronic numit demaror progresiv[114] şi care are la bază, în esență, un microcalculator specializat.

Elementul de execuție este în cazul de față un element de comutare de forță realizat cu ajutorul unui contactor[115]. Şi elementul de execuție trebuie proiectat[116] şi construit.

Schema bloc prezintă semnalele de intrare şi ieşire din fiecare bloc. În cazul de față se observă că atât automatul, cât şi elementul de execuție, au la intrare şi la ieşire semnale de tip contact.

[113] Un automat de pornire stea-triunghi, de exemplu.

[114] Soft starter (eng.)

[115] Comutatorul alimentării motorului electric poate avea şi alte tipuri de realizări, de exemplu folosind tiristoare.

[116] Din punct de vedere hardware şi software, dacă este cazul.

În continuare se va prezenta în special proiectarea automatului, elementului de execuţie, a sistemului de semnalizare şi comutare în diferite regimuri de lucru.

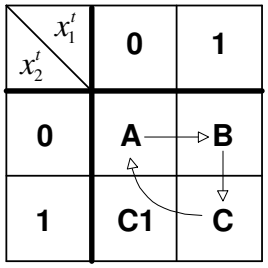


Fig. 5.46 Diagrama Karnaugh pentru codificarea adiacentă a poziţiilor reţelei Petri.

5.14.2. Caietul de sarcini al automatului

Caietul de sarcini. Algoritmul simplu de conducere al motorului este prezentat sub formă de grafcet în Fig. 5.45 a). Există trei etape şi trei tranziţii. Etapa A este activă iniţial şi prin convenţie, dacă nu este specificat altfel, se consideră că toate acţiunile, exprimate prin expresii logice, iau în această etapă valoarea *0*, adică valoarea logică *fals.* Deci şi motorul electric este oprit în această etapă.

5.14.3. Analiza structurală şi comportamentală

Analiza structurală. Să considerăm că există o reţea Petri echivalentă grafcetului. Structura sa este identică, iar din punct de vedere grafic pătratele (etapele) se înlocuiesc cu cercuri (poziţii). În etapa iniţială A se pun o singură marcă. Deoarece fiecare din cele trei tranziţii are un singur arc de intrare şi un singur arc de ieşire Reţeaua Petri este de tip maşină de stare. Graful asociat reţelei Petri este conex deoarece din orice nod[117] se poate ajunge în oricare alt nod, mai mult este tare conex deoarece această performanţă se realizează prin intermediul unor arce orientate. Teorema lui Commoner spune că dacă graful este tare conex atunci reţeaua este *viabilă*. Dacă marcajul iniţial are o singură marcă, cum este cazul aici, atunci reţeaua este viabilă şi *sigură*. Analizând structura reţelei se observă imediat că nu există conflicte structurale, deci reţeaua Petri este conformă şi echivalentă cu grafcetul. Toate concluziile pe care te obţinem lucrând cu reţeaua Petri sunt valabile şi pentru grafcet.

[117] Nodurile grafului sunt poziţiile. Laturile grafului sunt arcele orientate care leagă nodurile atunci când se ignoră tranziţiile.

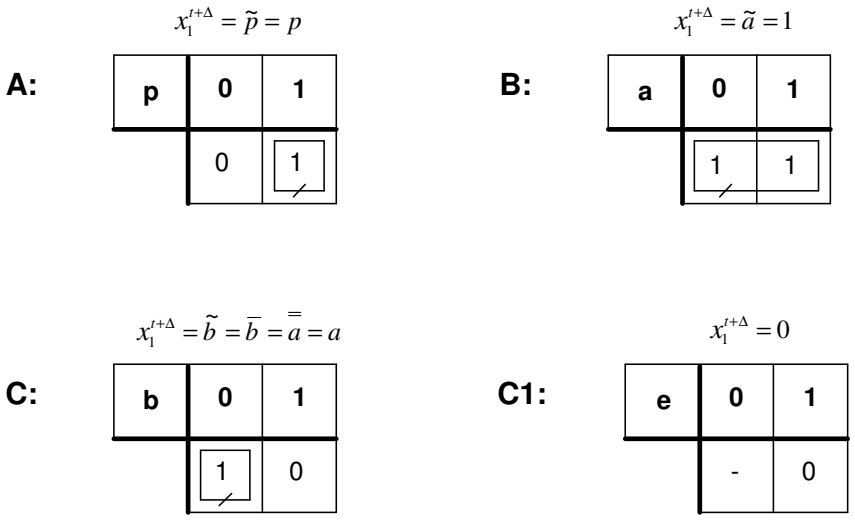


Fig. 5.47 Diagramele Karnaugh pentru stabilirea modului în care evoluează variabila de stare x_1 din fiecare poziţie a reţelei din Fig. 5.45 a).

Analiza comportamentală. Tranziţia (1), validată iniţial, se declanşează când se apasă butonul de pornire şi *p=1.* Devine activă etapa B şi motorul porneşte. La terminarea unei rotaţii contactul traductorului este acţionat de palpator şi *a=1.* Tranziţia (2) se declanşează şi devine activă etapa C. Această etapă nu este interpretată de nici o acţiune nouă. Motorul se roteşte în continuare. Rostul acestei etape este să memoreze că a apărut evenimentul specific terminării unei rotaţii. Cama rotindu-se încet, la un moment dat palpatorul nu mai este împins şi contactul *a* nu mai este acţionat. Expresia logică simplă care determină valoarea variabilei *b* va fi atunci egală cu unu (adevărat) şi tranziţia (3) se declanşează. Etapa A redevine activă. Există deci un ciclu repetitiv care coincide cu algoritmul dorit pentru automatul sistemului.

Analiza comportamentală trebuie să ia în considerare şi alte situaţii de funcţionare în afară de cea de bază. Ce se întâmplă dacă motorul se roteşte încet şi dorim să se oprească. Vedem că acest lucru nu este posibil. Conform caietului de sarcini motorul se roteşte întotdeauna până când se termină o rotaţie. Eventualele modificări ale caietului de sarcini se vor discuta o primă varianta a proiectului.

5.14.4. Sinteza automatului cu contacte şi relee.

Codificarea. Prima etapă a sintezei constă în codificarea binară a poziţiilor[118]. Având numai trei poziţii A, B şi C numărul de biţi necesari este egal cu doi. Într-adevăr, cu doi biţi se pot obţine $2^n=2^2=4$ coduri. Pentru a avea cât mai puţine relee şi pentru a nu se modifica la fiecare tranziţie decât un bit[119], se alege

[118] Etapelor, deoarece grafcetul şi reţeaua Petri sunt echivalente.

[119] Dacă s-ar modifica teoretic simultan mai mulţi biţi s-ar pune problema practică ce se întâmplă dacă totuşi un bit se modifică mai rapid decât altul. Pentru a se evita

codul binar Gray. Deoarece capetele de tabele ale diagramelor Karnaugh sunt scrise în cod Gray putem folosi această diagramă şi la codificare. Pentru aceasta se trec denumirile poziţiilor reţelei în pătrate adiacente[120] ale diagramei Karnaugh cu două variabile. Din acest motiv această codificare se mai numeşte *adiacentă*. Există mai multe soluţii de codificare. Una dintre ele se vede în Fig. 5.46. Deoarece pătratele diagramei în care se găsesc poziţiile A şi C ale reţelei nu sunt adiacente se introduce o poziţie nouă C1 în reţea. Această poziţie nu este interpretată de nici o acţiune. Rolul ei este de a asigura modificarea unui singur bit atunci când se trece de la poziţia C la poziţia iniţială A.

Codurile obţinute pentru cei doi biţi de stare x_1 şi x_2 sunt prezentate în Fig. 5.45 b). Se observă că la trecerea de la o poziţie la alta se modifică un singur bit. De exemplu, la declanşarea tranziţiei (2) poziţia C devine activă iar poziţia B inactivă. Bitul x_2 trece de la valoarea 0 la valoarea 1.

$x_1^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	p	1
1	0	a

a)

$x_2^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	0	a
1	0	1

b)

M^t

x_2^t \ x_1^t	0	1
0	0	1
1	-	1

c)

Fig. 5.48 Diagramele Karnaugh cu variabile înglobate ale variabilelor de stare şi de ieşire în cazul automatului pentru oprirea unui motor după o rotaţie.

Evoluţia variabilelor de stare. Fiecare poziţie este caracterizată de două variabile de stare: x_1 şi x_2. De exemplu, atunci când poziţia B este activă[121] x_1=1 şi x_2=0.

Acestea sunt valorile biţilor la momentul actual (prezent) de timp *t*. În viitorul apropiat, la momentul de timp imediat următor *t+Δ,* poziţia B poate să mai fie sau să nu mai fie activă. Dacă rămâne activă biţii îşi păstrează valoarea. Atunci când tranziţia (2) se declanşează biţii îşi modifică, eventual, valorile. Aceste valori noi depind de apariţia evenimentului *a* care interpretează tranziţia (2). De exemplu, pentru stabilirea relaţiei logice care determină valoarea bitului x_1 pentru poziţia B se completează diagrama Karnaugh corespunzătoare acestei poziţii din Fig. 5.47. Dacă evenimentul *a* nu apare, adică a=0, B rămâne activă şi x_1=1. Dacă evenimentul *a* apare, a=1, poziţia C devine activă şi valoarea lui x_1 rămâne egală tot cu 1. Făcând gruparea corespunzătoare rezultă relaţia logică. Se procedează la

---

rezolvarea problemei curselor dintre biţi, care u7neori pot deveni critice, se adoptă măsura de precauţie a modificării unui singur bit la declanşarea unei tranziţii.

[120] Pătratele adiacente au o latură comună în sensul definiţiei diagramei Karnaugh care prevede că laturile opuse ale diagramei sunt lipite.

[121] Marca se găseşte în ea.

fel pentru toate poziţiile şi pentru toţi biţii. Cu relaţiile obţinute se completează diagramele Karnaugh cu variabile înglobate din Fig. 5.48.

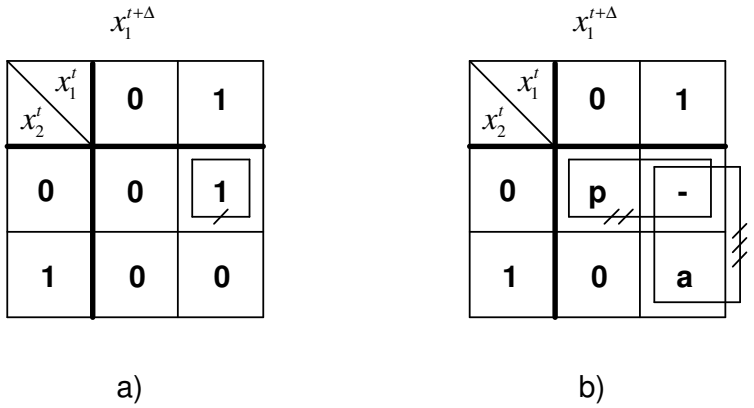


Fig. 5.49 Grupările pentru variabila de stare x_1 din Fig. 5.48 a).

Stabilirea relaţiei logice pe baza diagramei Karnaugh cu variabile înglobate se face în doi paşi.

- Pasul 1. Variabilele înglobate se fac egale cu zero şi diagrama se prelucrează în mod clasic. De exemplu, pentru variabila de stare x_1 definită în Fig. 5.48 se obţine diagrama din Fig. 5.49 a) cu o singură grupare.
- Pasul 2. În locul valorilor logice 1 se trec valori imposibile din punct de vedere tehnologic[122], marcate cu o liniuţă şi se fac grupări de pătrate care au aceiaşi variabilă înglobată. Pentru aceiaşi variabilă x_1 se obţine diagrama din Fig. 5.49 b) cu două grupări. Când se determină relaţia logică la fiecare grupare se ataşează şi variabila logică înglobată.

Pentru cele trei grupări din Fig. 5.49 a) şi b) rezultă relaţia logică (5.56) pentru x_1.

[122] Aceste situaţii nu există în realitate şi se poate considera o valoare logică 0 sau 1 astfel încât gruparea să fie cât mai mare şi deci simplificarea funcţiei logice să fie cât mai puternică.

$$\begin{aligned} x_1^{t+\Delta} &= \tilde{x}_1^t.\tilde{x}_2^t + p^t.\tilde{x}_1^t.\tilde{x}_2^t + a^t.\tilde{x}_1^t.\tilde{x}_2^t \\ x_1^{t+\Delta} &= x_1^t.\overline{x_2^t} + p^t.1.\overline{x_2^t} + a^t.x_1^t.1 \\ x_1^{t+\Delta} &= x_1^t.(\overline{x_2^t} + a^t) + p^t.\overline{x_2^t} \end{aligned} \qquad (5.56)$$

Diagramele Karnaugh din Fig. 5.48 b) şi c) permit calculul relaţiilor logice (5.57) şi (5.58)pentru x_2 şi M.

$$x_2^{t+\Delta} = x_1^t.(x_2^t + a^t) \qquad (5.57)$$

$$M^t = x_1^t \qquad (5.58)$$

5.14.5. Implementarea automatului cu contacte şi relee.

Alături de ecuaţiile de stare (5.56), (5.57) şi ieşire (5.58)se consideră relaţiile fizice ale releelor cu care implementăm cei doi biţi şi relaţiile de echivalenţă necesare.

$$\begin{aligned} k_1^{t+\Delta} &= K_1^t \\ k_2^{t+\Delta} &= K_2^t \\ x_1^t &= k_1^t \\ x_2^t &= k_2^t \end{aligned} \qquad (5.59)$$

Eliminând variabilele x_1, x_2, t şi Δ se obţin relaţiile logice (5.60) ale unui sistem logic combinaţionale format din două relee cu bobinele K_1 şi K_2 şi relaţia de ieşire.

$$\begin{aligned} K_1 &= k_1.(\overline{k}_2 + a) + p.\overline{k}_2 \\ K_2 &= k_1.(k_2 + a) \\ M &= k_1 \end{aligned} \qquad (5.60)$$

Schema electrică corespunzătoare relaţiilor logice (5.60) este prezentată în Fig. 5.50.

După implementare se face simularea automatului pentru a vedea dacă respectă caietul de sarcini din Fig. 5.45 a). La apăsarea butonului de pornire este acționat releul K1 din Fig. 5.50 care se automenține. În acelaşi timp porneşte motorul M care începe să se rotească. La terminarea unei rotații se acționează contactul *a* care provoacă atragerea releului K2 care se automenține. Motorul se roteşte în continuare. În momentul în care cama nu mai împinge palpatorul se deschide contactul *a* Ca urmare se realizează strict următoarea secvență de acțiuni: 1) releul K1 nu mai este acționat, 2) se opreşte motorul M şi releul K2 nu mai este acționat. Se ajunge în starea inițială şi se poate relua ciclul dacă se acționează butonul mde pornire *p.*

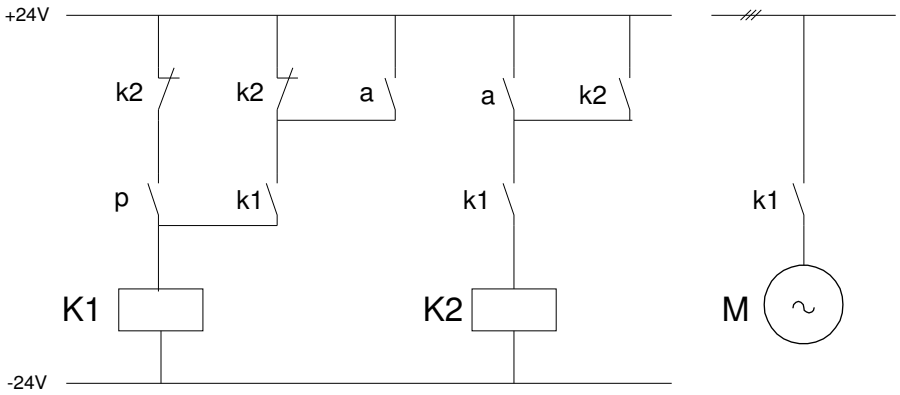


Fig. 5.50 Rezultatul sintezei şi implementării cu contacte şi relee a automatului pentru oprirea unui motor după o rotație.

Rezultatul sintezei şi implementării automatului din Fig. 5.50 nu este schema electrică desfăşurată finală. Pentru obținerea acesteia mai trebuiesc efectuate multe operații de proiectare. Să menționăm acum câteva dintre cele mai importante.

- Se introduce un sistem de alimentare a automatului.
- Se introduce un sistem de protecție a motorului, rețelei electrice de alimentare şi echipamentului de automatizare. Se foloseşte metoda exemplelor tip.
- Se introduce un sistem de semnalizare a principalelor stări, acțiuni şi evenimente care apar în funcționarea sistemului. Se foloseşte metoda exemplelor tip.
- Se introduce un sistem de deparazitare a perturbațiilor pe care le poate provoca în timpul funcționării automatul. Se foloseşte metoda exemplelor tip.
- Se dimensionează şi se aleg din catalog componentele automatului.
- Se dimensionează cutia în care se montează componentele automatului.
- Se întocmeşte documentația necesară pentru realizarea tehnologică a automatului.

5.15. Proiectarea automatelor implementate cu bistabile

Se foloseşte în continuare exemplul automatului pentru oprirea motorului după o rotaţie.

5.15.1. Sinteza şi implementarea automatului cu bistabile tip D

Codificarea. Deoarece bistabilele master – slave îşi schimba starea numai la momente exacte de timp, determinate de impulsurile de ceas, nu există restricţii la modul de codificare. Cu alte cuvinte se pot modifica mai mulţi biţi simultan la trecerea de la o poziţie la alta. Pentru codificarea din Fig. 5.45 b) nu mai este necesară introducerea unei poziţii suplimentare şi se obţine cazul din Fig. 5.51 a). În cazul acesta la trecerea de la poziţia cu codul 11 la poziţia cu codul 00 se modifică simultan ambii biţi.

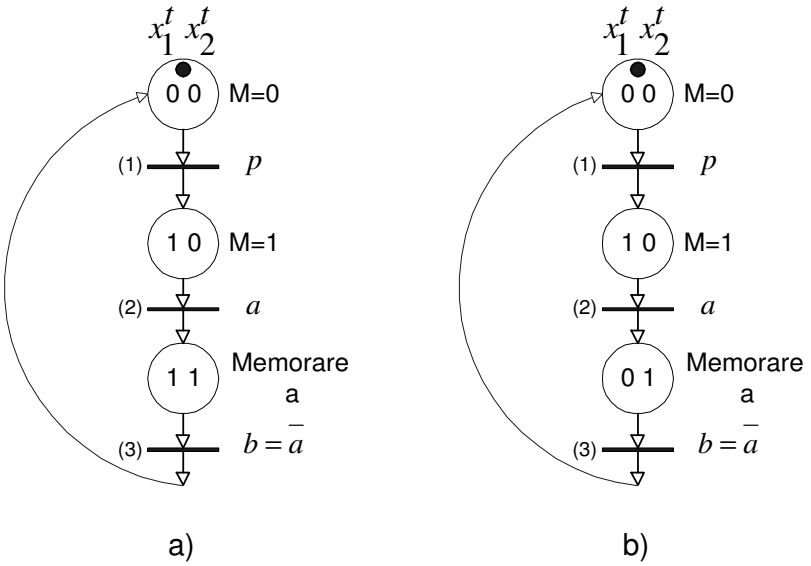


Fig. 5.51 Codificări neadiacente la sinteza automatului pentru oprirea unui motor după o rotaţie.

Pentru codificarea din Fig. 5.51 b) se modifică doi biţi când se declanşează tranziţia (2), iar în cazul codificării din Fig. 5.51 c), unde se folosesc trei biţi, se modifică doi dintre ei la declanşarea fiecărei tranziţii. Ultima codificare are avantajul că fiecare poziţie este caracterizată de un singur bit. În acest caz este mai uşor de înţeles funcţionarea şi de realizat sinteza şi implementarea. Preţul plătit pentru aceasta este un bistabil pentru fiecare poziţie.

Diagramele Karnaugh pentru biţii de stare şi variabila de ieşire ai reţelei din Fig. 5.51 a) sunt prezentate în Fig. 5.52. Relaţiile logice se stabilesc la fel ca în cazul sintezei cu contacte şi relee folosind metoda diagramelor Karnaugh cu variabile înglobate. Rezultă relaţiile (5.61). Să vedem problemele care apar în acest caz datorită modificării nesimultane a doi biţi în cazul implementării cu contacte şi relee.

$x_1^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	p	1
1	0	a

a)

$x_2^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	0	a
1	0	a

b)

M^t

x_2^t \ x_1^t	0	1
0	0	1
1	-	1

c)

Fig. 5.52 Diagramele Karnaugh pentru sinteza automatului corespunzător reţelei din Fig. 5.51 a).

$$\begin{aligned} x_1^{t+\Delta} &= x_1^t.(\overline{x_2^t} + a^t) + p^t.\overline{x_2^t} \\ x_2^{t+\Delta} &= a.x_1^t \\ M^t &= x_1^t \end{aligned} \qquad (5.61)$$

Adăugăm relaţiilor logice (5.61) relaţiile fizice ale releelor şi relaţiile de echivalenţă (5.59). După eliminarea variabilelor de stare x_1, x_2, a timpului t şi variaţiei de timp Δ rezultă sistemul de relaţii logice (5.62) care poate fi implementat cu contacte şi relee ca în Fig. 5.53.

$$\begin{aligned} K_1 &= k_1.(\overline{k}_2 + a) + p.\overline{k}_2 \\ K_2 &= k_1.a \\ M &= k_1 \end{aligned} \qquad (5.62)$$

Schema din Fig. 5.53 diferă de cea din Fig. 5.50 numai prin faptul că releul K2 nu are un contact de automenţinere. Schema este mai simplă, dar după activarea poziţiei caracterizate de codul 11 datorită acţionării contactului *a* poate apare următoarea situaţie. Ambele relee K1 şi K2 sunt acţionate prin circuite identice formate din contactele închise *a* şi *k1*. Când cama nu mai împinge palpatorul la terminarea unei rotaţii complete, Fig. 5.43, contactul *a* se deschide şi releele K1 şi K2 ar trebui să se dezactiveze *simultan*. Dacă din motive *fizice aleatorii* releul K2 se dezactivează mai încet decât K1, acesta din urmă se automenţine prin circuitul format din contactele închise k2 şi k1. Releul K1 rămânând acţionat motorul M merge în continuare deşi ar trebui să se oprească

deoarece s-a terminat o rotație. Din această cauză schema din Fig. 5.53 nu este acceptabilă în practică şi se foloseşte schema din Fig. 5.50 rezultată în urma unei codificări adiacente care asigură că nici o declanşare de tranziție nu va modifica simultan două variabile de stare.

Pentru implementarea cu bistabile master – slave tip D a relațiilor logice (5.61) obținute în urma unei codificări neadiacente se folosesc relațiile fizice ale bistabilelor D şi relațiile de echivalență a variabilelor de stare cu ieşirile bistabilelor (5.63).

$$\begin{aligned} Q_1^{t+\Delta} &= D_1^t \\ Q_2^{t+\Delta} &= D_2^t \\ x_1^t &= Q_1^t \\ x_2^t &= Q_2^t \end{aligned} \qquad (5.63)$$

După eliminarea variabilelor de stare x_1, x_2, a timpului t şi variației de timp Δ se obțin relațiile (5.64) pe baza cărora se elaborează schema electrică din Fig. 5.54.

$$\begin{aligned} D_1 &= Q_1.(\overline{Q}_2 + a) + p.\overline{Q}_2 \\ D_2 &= Q_1.a \\ M &= Q_1 \end{aligned} \qquad (5.64)$$

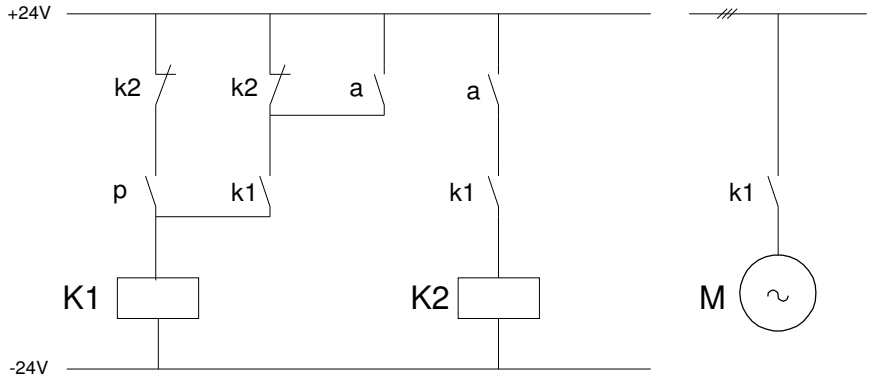


Fig. 5.53 Schema electrică în cazul codificării neadiacente din Fig. 5.51 a).

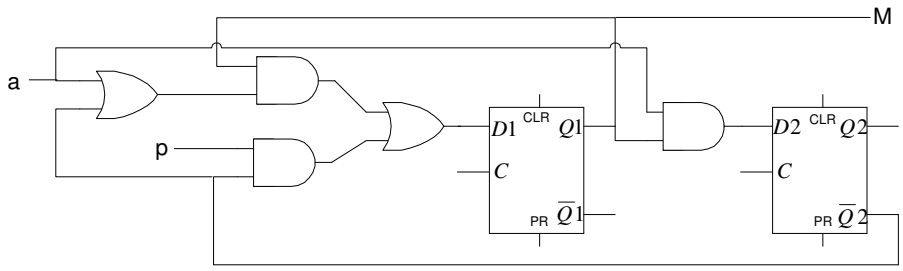


Fig. 5.54 Schema electrică a automatului pentru oprirea unui motor după o rotație în varianta implementată cu bistabile master – slave tip D.

5.15.2. Sinteza şi implementarea automatului cu bistabile JKC.

O altă variantă de codificare neadiacentă este prezentată în Fig. 5.51 b). De data aceasta se modifică doi biți la declanşarea tranziției (2).

$x_1^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	p	$\overline{a}$
1	0	0

a)

$x_2^{t+\Delta}$

x_2^t \ x_1^t	0	1
0	0	a
1	a	0

b)

M^t

x_2^t \ x_1^t	0	1
0	0	1
1	1	-

c)

Fig. 5.55 Diagramele Karnaugh pentru rețeaua din Fig. 5.51 b).

Evoluția variabilelor de stare este prezentată în Fig. 5.55 a) şi b) iar variabila de ieşire în funcție de variabilele de stare este determinată de diagrama Karnaugh din Fig. 5.55 c). Relațiile logice corespunzătoare sunt (5.65).

$$\begin{aligned} x_1^{t+\Delta} &= p^t.\overline{x_2^t}.\overline{x_1^t} + \overline{a}.\overline{x_2^t}.x_1^t \\ x_2^{t+\Delta} &= a.x_1^t.\overline{x_2^t} + a.\overline{x_1^t}.x_2^t \\ M^t &= x_1^t + x_2^t \end{aligned} \qquad (5.65)$$

Din forma relațiilor fizice rezultă că se pot folosi la implementare bistabile JKC[123]. În (5.66) se prezintă relațiile fizice ale acestor bistabile şi relațiile de echivalență cu variabilele de stare.

[123] Bineînțeles că se pot folosi şi bistabile D, la fel ca în paragraful precedent.

$$
\begin{aligned}
Q_1^{t+\Delta} &= J_1^t.\overline{Q_1^t} + \overline{K_1^t}.Q_1^t \\
Q_2^{t+\Delta} &= J_2^t.\overline{Q_2^t} + \overline{K_2^t}.Q_2^t \\
x_1^t &= Q_1^t \\
x_2^t &= Q_2^t
\end{aligned}
\qquad (5.66)
$$

Eliminând variabilele x_1, x_2, t şi Δ rezultă relațiile (5.67) care pot fi implementate ca în Fig. 5.56.

$$
\begin{aligned}
&J_1 = p.\overline{Q_2} \\
&\overline{K_1} = \overline{a}.\overline{Q_2}\ ,\ K_1 = a + Q_2 \\
&J_2 = a.Q_1 \\
&\overline{K_2} = a.\overline{Q_1}\ ,\ K_2 = \overline{a} + Q_1
\end{aligned}
\qquad (5.67)
$$

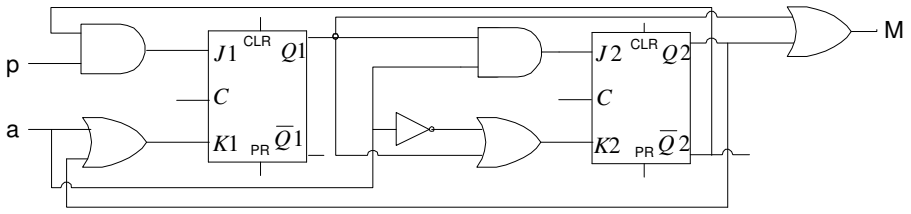


Fig. 5.56 Schema electrică a automatului pentru oprirea după o rotație în varianta implementată cu bistabile JKC master – slave.

5.15.3. Problema inițializării sistemului discret logic.

Problema inițializării, adică a stabilirii stării inițiale a sistemului[124], este adeseori tratată superficial deşi poate avea implicații mari asupra performanțelor sistemului.

[124] Stabilirea etapelor inițiale la grafcet sau a marcajului inițial la rețeaua Petri.

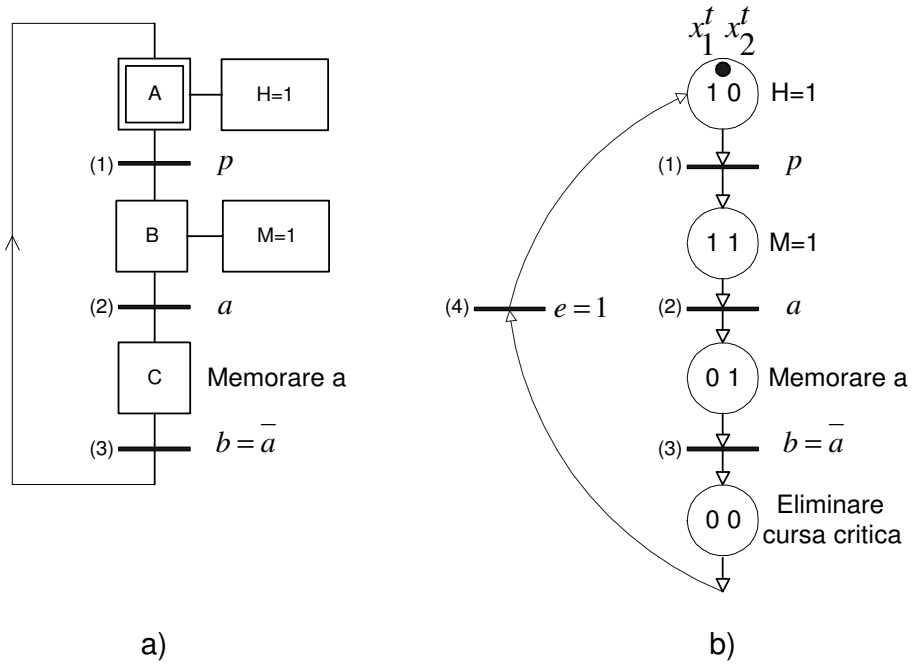


Fig. 5.57 Caietul de sarcini şi o codificare adiacentă pentru automatul care opreşte un motor după o rotaţie cu semnalizarea iniţializării.

La întocmirea caietului de sarcini din Fig. 5.45 pentru automatul care opreşte un motor electric după o rotaţie s-a considerat că etapa iniţială corespunde cu starea în care motorul este oprit, adică nealimentat. Dacă se acţionează butonul de pornire *p* motorul execută o rotaţie. Cu acest caiet de sarcini, în situaţia în care motorul se opreşte accidental, de exemplu la întreruperea alimentării cu energie electrică, la reluarea alimentării el ajunge în etapa A, Fig. 5.45 a), adică sistemul este iniţializat. La acţionarea butonului de pornire motorul nu mai execută însă o rotaţie completă! Acest lucru poate să fie foarte supărător în unele aplicaţii. Pentru eliminarea acestui inconvenient stabilim prin caietul de sarcini că în starea iniţială motorul trebuie să se găsească exact în situaţia din Fig. 5.43, adică într-o situaţie identică cu cea în care ajunge după terminarea unei rotaţii. Dacă se găseşte în starea iniţială, adică sistemul este iniţializat, lampa H semnalizează.

Pentru evitarea coincidenţii stării iniţiale cu starea oprit (nealimentat), la implementarea cu contacte şi releu, se codifică diferit de codul 00 poziţia iniţială a reţelei Petri Fig. 5.57 b) echivalente caietului de sarcini. O variantă de codificare adiacentă este cea prezentată în Fig. 5.58. Sinteza automatului se face la fel ca în paragraful 5.14.4. Se întocmesc diagramele Karnaugh din Fig. 5.59 pentru evoluţia variabilelor de stare şi pentru determinarea variabilelor de ieşire, se fac grupări şi se determină relaţiile logice (5.68) după metoda variabilelor înglobate, dacă se consideră relaţiile fizice şi de echivalenţă (5.59)

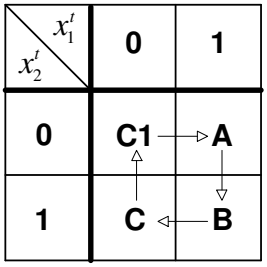


Fig. 5.58 Codificarea adiacentă a automatului cu inițializare.

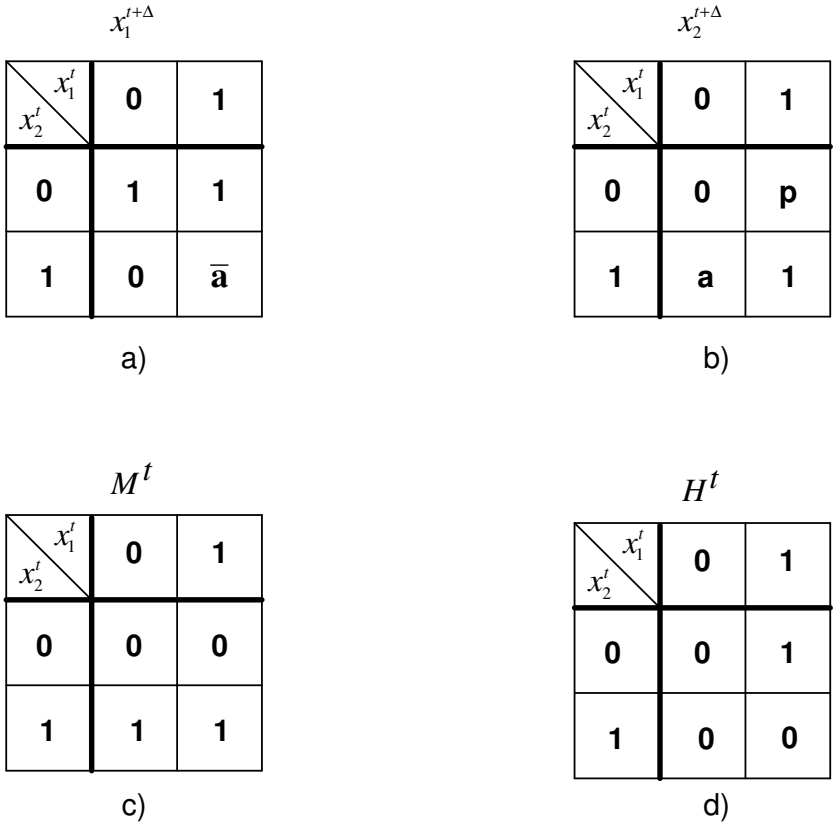


Fig. 5.59 Diagramele Karnaugh pentru evoluția variabilelor de stare şi pentru variabilele de ieşire în cazul automatului cu inițializare care opreşte un motor după o rotație.

Schema electrică corespunzătoare relațiilor (5.68) este prezentată în Fig. 5.60.

$$\begin{aligned} K_1 &= \overline{k}_2 + \overline{a}.k_1 \\ K_2 &= p.k_1 + (k_1 + a).k_2 \\ M &= k_2 \\ H &= k_1.\overline{k_2} \end{aligned} \qquad (5.68)$$

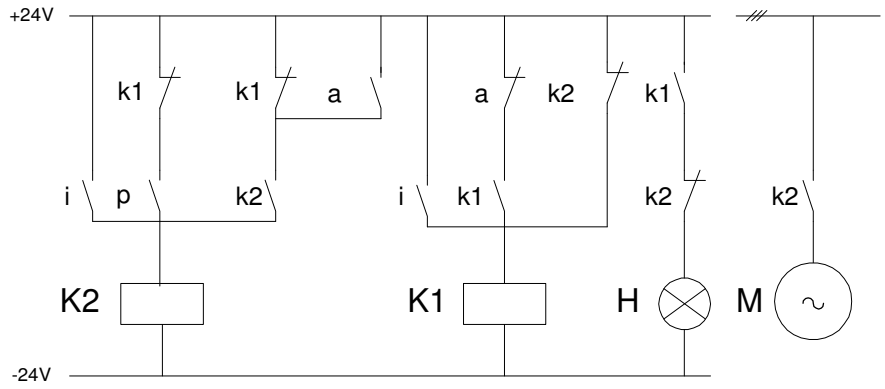


Fig. 5.60 Schema electrică a automatului cu inițializare pentru oprirea unui motor electric după o rotație.

Inițializarea sistemului se face prin apăsarea butonului i din Fig. 5.60. Se observă că în felul acesta sunt acționate ambele relee şi sistemul ajunge în poziția B din Fig. 5.57 b) codificată cu x1=1 şi x2=1. Motorul începe să se rotească, indiferent în ce poziție se află, execută o mişcare de rotație până când contactul a se deschide şi se închide la loc. După aceasta motorul se opreşte în poziția inițială A codificată cu x1=1 şi x2=0, iar lampa H se aprinde semnalizând inițializarea sistemului. După inițializare sistemul îşi poate relua ciclul normal de funcționare prin acționarea butonului de pornire.

La implementarea cu circuite bistabile inițializarea are aceiaşi importanță. Ea se face aplicând semnalele corespunzătoare pe bornele de ştergere CLR şi de aducere pe poziția unu logic PR ale bistabilelor, Fig. 5.54 şi Fig. 5.56. În acest caz codul 00 nu mai coincide cu situația în care bistabilele au fost puse în funcțiune prin alimentare cu energie electrică, aşa cum era cazul la implementarea cu contacte şi relee. Din această cauză inițializarea trebuie făcută întotdeauna, indiferent de semnificația poziției inițiale.

5.16. Proiectarea automatelor implementate cu APL

În continuare se prezintă mai multe metode[125] de proiectare a automatelor cu evenimente discrete care sunt implementate cu automate programabile logice şi se foloseşte acelaşi exemplu de automat pentru oprirea unui moto după o rotație.

5.16.1. Programarea automatelor programabile logice

Operațiunile logice de bază pe care le realizează un automat programabil sunt prezentate pe scurt în continuare folosind schema bloc din Fig. 5.61.

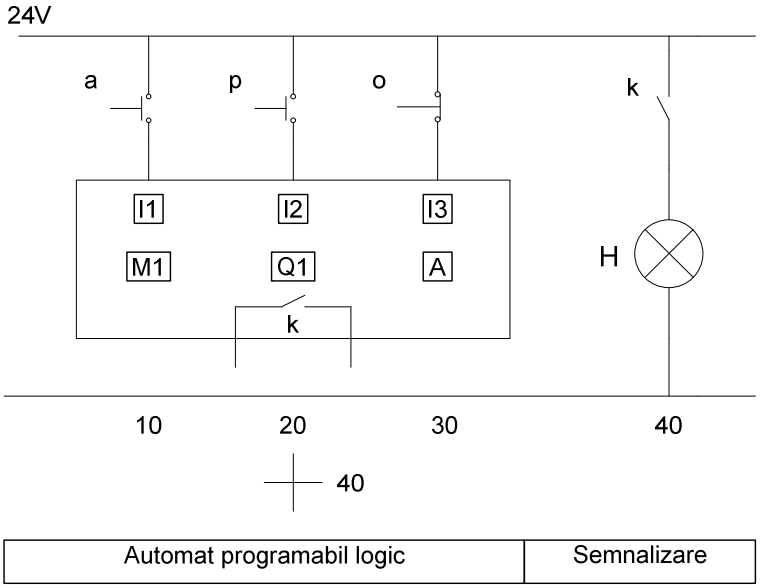


Fig. 5.61 Schema bloc a unui automat programabil logic

Un automat programabil logic are mai multe celule de memorie specializate, fiecare dintre ele caracterizate printr-o adresă[126]. Câteva tipuri de celule de memorie mai importante sunt următoarele:

- Intrările, de exemplu I1, I2 şi I3. La aceste intrări se conectează dispozitive care furnizează +24V pentru unu logic şi zero volți pentru unu logic. La intrare I3 este conectat un buton de oprire. El este normal închis , astfel în cât ruperea circuitului să echivaleze cu comanda oprit.
- Ieşirile. În Fig. 5.61 apare numai ieşirea Q1. Valoarea unu logic în celula de ieşire acționează un releu cu contactul k.
- Memoriile, În aceste celule, de exemplu M1, se memorează rezultatele intermediare.

[125] Standardizate în IEC 61131-3

[126] Adresele sunt numerotate începând cu zero în standardul IEC 61131-3 şi la unele AP cum ar fi cele produse de Moeller, sau începând cu unu în cazul AP Trilogi.

- Acumulatorul A. Există un singur acumulator în care se realizează toate operațiile logice. Un operand trebuie să se găsească întotdeauna în acumulator. Rezultatul operației logice apare tot în acumulator.

Câteva instrucțiuni apar în Tab 5.1. Dacă după operatorul instrucțiunii apare modificatorul N atunci operandul x este negat. De exemplu, ***LDN x*** încarcă în acumulatorul A valoarea negată a lui x. Câteva aplicații ale instrucțiunilor în scurte secvențe de program tip IL şi LD sunt prezentate în Tab 5.2

Tab 5.1 Lista instrucțiunilor principale a unui automat programabil logic.

IEC 61131	*Moeller*	*Modificator operand*	*Actiunea asupra operand x. Acesta este negat daca apare modificatorul N*
LD x	***L x***	***N***	***Load x →A***
ST x	***= x***	***N***	***Store A →x***
AND x	***A x***	***N***	***A AND x →A***
OR x	***O x***	***N***	***A OR x →A***
S x	***S x***		***Set x=1 if A=1***
R x	***R x***		***Reset x=0 if A=1***
XOR x	***XO x***		***A XOR x →A***

Două medii de dezvoltare a programelor sunt prezentate în ultimul capitol.

Tab 5.2 Secvențe tipice de programare

Nr.	***Program IL***	***Program LD***	***Relația logica***
1	***L I1*** ***A I2*** ***= Q1***	***I1 I2 Q1*** ***|------| |-------| |-----------( )---|***	***H=a.p*** ***Elementul SI***
2	***L I1*** ***O I2*** ***= Q1***	***I1 Q1*** ***|------| |----|----------( )---|*** ***| I2 |*** ***|------| |----|***	***H=a+p*** ***Elementul SAU***
3	***LN I1*** ***=Q1***	***I1 Q1*** ***|------| / |----------( )---|***	***H=/a*** ***Elementul NU cu modificatorul N***
4	***L I1*** ***A I2*** ***=N Q1***	***I1 I2 M1*** ***|------| |-------| |-----------( )---|*** ***| M1 Q1 |***	***H=/(a.p)*** ***Elementul SI-NU***

		|------| / |----------( )--|	
5	***L I1*** ***O I2*** ***=N Q1***	***I1 M1*** ***|------| |----|----------( )---|*** ***| I2 | |*** ***|------| |----| |*** ***| M1 Q1 |*** ***|------| / |----------( )---|***	***H=/(a+p)*** ***Elementul SAU-NU***
6	***L I1*** ***A I3*** ***= M1*** ***L I2*** ***AN I3*** ***O M1*** ***= Q1***	***I1 I3 Q1*** ***|------| |----| |----|----------( )---|*** ***| I2 I3 |*** ***|------| |----| / |----|***	***H=a./o+p.o*** ***Folosirea memoriei tampon M***
8	***L I2*** ***O Q1*** ***A I3*** ***=Q1***	***I3 I2 Q1*** ***|------| |----|------| |---|---( )---|*** ***| | Q1 | | |*** ***| |------| |---| |***	***H= /o.(p+k)*** ***Auto-mentinere cu prioritate la oprire. Oprirea o nu este negata pentru siguranța la avarie[127]***
9	***L I2*** ***S Q1*** ***L I1*** ***R Q1***	***I2 SQ1*** ***|------| |----------( )---|*** ***| I1 RQ1 |*** ***|------| / |----------( )---|***	***Bistabil SR cu prioritate la oprire*** ***p = Set*** ***a = Reset***
10	***L I1*** ***R Q1*** ***L I2*** ***S Q1***	***I1 RQ1*** ***|------| / |----------( )---|*** ***I2 SQ1*** ***|------| / |----------( )---|***	***Bistabil SR cu prioritate la pornire*** ***p = Set*** ***a = Reset***

[127] „Fail – safety” la intreruperea firului

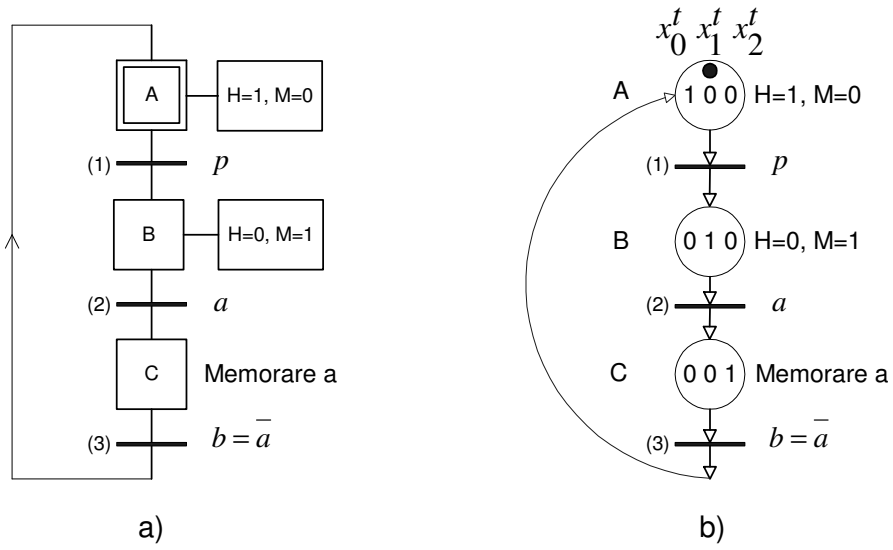


Fig. 5.62 Codificarea distribuită (1 din n) a caietului de sarcini cu inițializare pentru automatul care opreşte un motor după o rotație.

5.16.2. Metoda Grafcet[128]

Caietul de sarcini Grafcet din Fig. 5.62 a) poate fi implementat direct la unele automate programabile folosind o interfață grafică. Metoda este foarte utilă în cazul automatelor complexe.

5.16.3. Metoda listei de instrucțiuni[129]

Se foloseşte rețeaua Petri conformă care descrie funcționarea automatului. Pentru exemplul considerat aceasta este prezentată în Fig. 5.62 b). Metoda de proiectare are următoarele etape:

Etapa 1. Codificarea locațiilor cu ajutorul codului distribuit (1 din n). Rezultatul apare în Fig. 5.62 b). Se observă că fiecărei locații îi corespunde un singur bit egal cu unu. Metoda de proiectare se bazează pe observația că acest bit poate fi implementat cu ajutorul unui automat elementar cu prioritate la pornire. Condiția de setare coincide cu condiția de activare a locației iar condiția de resetare coincide cu condiția de dezactivare a locației.

Etapa 2. Calculul condițiilor de setare şi resetare a biților corespunzători fiecărei etape. Setarea coincide cu activarea locației care are loc dacă toate locațiile precedente sunt active *şi* evenimentele care interpretează tranzițiile de la aceste locații au apărut, adică au valoarea logică unu. Resetarea are loc dacă locația respectivă este activă *şi* evenimentele care declanşează tranzițiile

[128] Sequential Flow Chart (SFC)
[129] Instruction List (IL)

posterioare tranziției au apărut. Exemplul studiat este mai simplu pentru că fiecare locație are câte o singură tranziție precedentă şi o singură tranziție posterioară

$$n=0 \quad s_0 = x_2 \bar{a} \quad r_0 = x_0 p \tag{5.69}$$

$$n=1 \quad s_1 = x_0 p \quad r_1 = x_1 a \tag{5.70}$$

$$n=2 \quad s_2 = x_1 a \quad r_2 = x_2 \bar{a} \tag{5.71}$$

Etapa 3. Determinarea relațiilor logice pentru locațiile active inițial cu ajutorul următoarei formule care provine din expresia automatului elementar cu prioritate la pornire:

$$x_n^{t+\Delta} = s_n^t + \overline{r_n^t} x_n^t + i \tag{5.72}$$

Se calculează simplificările posibile.

În exemplul nostru este activă inițial locația A caracterizată de bitul cu n=0. Înlocuind condițiile (5.69) în (5.72) se obține:

$$x_0^{t+\Delta} = x_2^t \bar{a} + \overline{(x_0^t p)} x_0^t + i = x_2^t \bar{a} + (\overline{x_0^t} + \overline{p^t}) x_0^t + i = x_2^t \bar{a} + \overline{p^t} x_0^t + i \tag{5.73}$$

Etapa 4. Determinarea relațiilor logice pentru locațiile inactive la momentul de timp inițial. Se folosește tot o formulă derivată din relația logică a automatului elementar cu prioritate la pornire:

$$x_n^{t+\Delta} = (s_n^t + \overline{r_n^t} x_n^t) \bar{i} \tag{5.74}$$

Pentru automatul din Fig. 5.62 b) locațiile B şi C nu sunt active inițial şi aplicând formula precedentă pentru condițiile (5.70) şi (5.71) se obține:

$$x_1^{t+\Delta} = (x_0^t p^t + \bar{a} x_1^t) \bar{i} \tag{5.75}$$

$$x_2^{t+\Delta} = (x_1^t + x_2^t) a^t \bar{i} \tag{5.76}$$

Etapa 5. Determinarea relațiilor logice dintre stările x şi ieşirile y ale automatului la momentul de timp t. Aceste relații rezultă din rețeaua Petri care specifică pentru fiecare locație, caracterizată prin anumite valori ale variabilelor de stare x, care valorile variabilelor de ieşire y. De exemplu, pentru rețeaua Petri din Fig. 5.62 b) variabila de ieşire y_m care comandă motorul M are valoarea 1 numai pentru locațiile B şi C. Diagrama Karnaugh corespunzătoare şi relația logică stabilită[130] sunt prezentate în Fig. 5.63

[130] Există tentația ca în cazul unor relații logice simple, cum este cazul exemplului prezentat, să se deducă direct relația logică fără ajutorul diagramei Karnaugh. Este o metodă

$y_m^t = x_1^t + x_2^t$

x_2^t \ $x_0^t x_1^t$	00	01	11	10
0	0	1	-	0
1	1	-	-	-

Fig. 5.63 Determinarea relației logice pentru variabila de ieşire y_m care comandă motorul

$y_h^t = x_0^t$

x_2^t \ $x_0^t x_1^t$	00	01	11	10
0	0	0	-	1
1	0	-	-	-

Fig. 5.64 Determinarea relației logice pentru variabila de ieşire y_h care comandă lampa de semnalizare

Din aceiaşi rețea Petri rezultă că lampa H semnalizează numai în locația A şi deci numai atunci variabila de ieşire corespunzătoare y_h are valoarea 1. Relația logică corespunzătoare determinată cu ajutorul diagramei Karnaugh este prezentată în Fig. 5.64.

Etapa 6. Elaborarea tabelului de configurare. Se stabileşte o corespondență între variabilele relațiilor logice şi denumirile elementelor componente ale automatului programabil: intrări, ieşiri, memorii (relee), timere (relee de timp), etc. Pentru exemplul considerat acestea apar în Tab 5.3 şi Tab 5.4.

---

greşită pentru că permite ignorarea anumitor situații care sunt puse însă în evidență de către diagrama Karnaugh. În exemplul prezentat acesta este cazul cu situația în care toate variabilele de stare sunt egale cu zero, situație care un apare în rețeaua Petri. Diagrama Karnaugh ne-a silit să precizăm valorile de ieşire în acest caz. Am hotărât, de exemplu, că în acest caz lampa de semnalizare H nu este aprinsă.

Tab 5.3 Configurare APL pentru intrări şi ieşiri

i	p	a	y_m	y_h
I0.0	I0.1	I0.2	Q0.1	Q0.0
M0.0	M0.1	M0.2	M0.14	M0.15

Tab 5.4 Configurare APL pentru stări şi memorii de lucru

x_0^t	x_1^t	x_2^t	$x_0^{t+\Delta}$	$x_1^{t+\Delta}$	$x_2^{t+\Delta}$	*tampon1*	*tampon2*
M0.3	M0.4	M0.5	M0.6	M0.7	M0.8	M0.9	M0.10
Q0.3	Q0.4	Q0.5					

Etapa 7. Elaborarea programului de funcţionare a automatului programabil logic sub forma unei liste de instrucţiuni[131].

Pentru exemplul studiat lista de instrucţiuni corespunzătoare automatului programabil logic Klockner Moeller PS3 este prezentată în Tab 5.5. Se observă că programul are următoarele secţiuni:

- ✓ Achiziţia intrărilor (Input scan)
- ✓ Calculul relaţiilor logice (Logic scan)
- ✓ Actualizarea variabilelor de stare
- ✓ Calculul variabilelor de ieşire
- ✓ Furnizarea ieşirilor (Output scan)
- ✓ Operaţiuni de semnalizare sau testare

Tab 5.5 Programul sub formă de listă de instrucţiuni

Adresa	***Instrucţiunea***	***Comentariu***
000	LI0.0	Achiziţia intrărilor
001	=M0.0	
002	LI0.1	
003	=M0.1	
004	LI0.2	
005	M0.2	
006	LM0.5	Calcul $x_0^{t+\Delta}$
007	ANM0.2	
008	=M0.9	
009	LNM0.1	
010	AM0.3	
011	=M0.10	
012	LM0.9	
013	OM0.10	
014	OM0.0	
015	=M0.6	
016	LM0.3	Calcul $x_1^{t+\Delta}$
017	AM0.11	
018	=M0.9	

[131] Instruction List - IL

```
019        LNM0.2
020        AM0.4
021        =M0.10
022        LM0.9
023        OM0.10
024        ANM0.0
025        =M0.7
026        LM0.4        Calcul x2 t+Δ
027        AM0.2
028        =M0.9
029        LM0.2
030        AM0.5
031        =M0.10
032        LM0.9
033        OM0.10
034        ANM0.0
035        =M0.8
036        LM0.6        Actualizare stări
037        =M0.3        xn t+Δ --> xn t
038        LM0.7
039        =M0.4
040        LM0.8
041        =M0.5
042        LM0.4        Calcul ieşiri yn t
043        OM0.5
044        =M0.14
045        LM0.3
046        =M0.15
047        LM0.14       Furnizare ieşiri yn t
048        =Q0.1
049        LM0.15
050        =Q0.0
051        LM0.3        Semnalizare stări
052        =Q0.3
053        LM0.4
054        =Q0.4
055        LM0.5
056        =Q0.5
```

5.16.4. Metoda schemei desfăşurate[132]

Programul automatului programabil logic poate fi elaborat şi sub forma unei scheme desfăşurate electrice. Metoda este aceiaşi cu cea prezentată în paragraful 5.14.5 dar de data aceasta se folosesc relaţiile logice elaborate stabilite prin metoda prezentată în paragraful 5.16.3. Schema desfăşurată obţinută pentru APL Klockner Moeller Easy 412 este prezentată în Fig. 5.66 conform standardului

[132] Ladder Diagram (LD)

DIN şi IEC iar tabelul de configurare corespunzător în Tab 5.3. Un exemplu de program sub formă de schemă desfăşurată desenată conform standardului american ANSI/CSA se prezintă în paragraful 10.5.

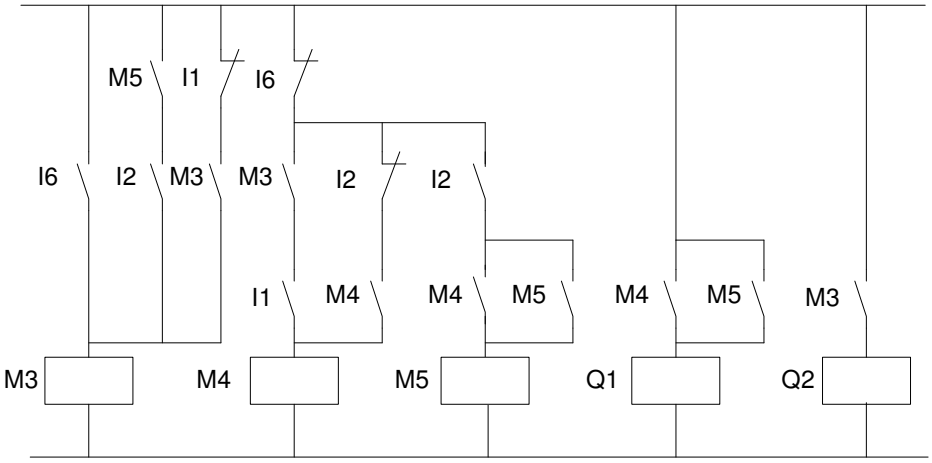


Fig. 5.65 Programul sub formă de schemă desfăşurată pentru automatul Klockner Moeller Easy 412 care opreşte un motor electric după o rotaţie

Tab 5.6 Tabelul de configurare pentru APL tip Klockner Moeller Easy 412

i	p	a	y_m	y_h	x_0^t	x_1^t	x_2^t
I6	I1	I2	Q1	Q4	M3	M4	M5

5.17. Transformarea APL în automat cu contacte şi relee

Metodele de sinteză şi a automatelor implementate cu contacte şi relee sau cu bistabile care au fost prezentate în paragrafele precedente au dezavantajul că conduc la sisteme discrete cu o funcţionare care poate fi înţeleasă mai greu şi prin urmare exploatarea şi întreţinerea lor este mai dificilă. Aceasta se datorează faptului că fiecare stare a automatului este determinată de un număr relativ mare[133] de relee sau bistabile. În plus, metoda de sinteză şi implementare bazată pe metoda diagramelor Karnaugh cu variabile înglobate nu este prea simplă. Este drept că în acest fel rezultă automate cu un număr mic de relee sau bistabile.

O alternativă la metodele clasice de proiectare o constituie transformarea schemei desfăşurate care reprezintă programul pentru un APL într-o schemă desfăşurată electrică. Transformarea este imediată şi efectul ei asupra schemei din *Fig. 5.65* se observă în Fig. 5.66. Elementele componente ale schemei reprezintă de data aceasta dispozitive fizice: bobine, contacte, butoane, etc. În afară de aceasta mai există o schimbare importantă. Regulile de evoluţie a unei reţele Petri, de exemplu cea din Fig. 5.62 b), precizează că la declanşarea unei tranziţii toate locaţiile precedente se dezactivează iar locaţiile posterioare se activează *simultan.* Deoarece condiţia de simultaneitate nu poate fi îndeplinită la implementarea cu

[133] În exemplul simplu al automatului pentru oprirea unui motor după o rotaţie se folosesc două relee sau bistabile pentru caracterizarea fiecărei stări (etape, poziţii). La automatele de complexitate medie se folosesc frecvent trei sau patru relee sau bistabile.

contacte şi relee este necesar ca mai întâi să se activeze locaţia posterioară şi apoi să se dezactiveze locaţia precedentă. Aceasta se poate realiza impunând ca condiţia de resetare a unei locaţii să fie identică cu activarea locaţiei posterioare. Relaţiile logice (5.69), (5.70) şi (5.71) se transformă astfel în relaţiile logice (5.77), (5.78) şi (5.79) în care numai condiţiile de resetare sunt modificate.

$$n=0 \quad s_0 = x_2 \bar{a} \quad r_0 = x_1 \tag{5.77}$$

$$n=1 \quad s_1 = x_0 p \quad r_1 = x_2 \tag{5.78}$$

$$n=2 \quad s_2 = x_1 a \quad r_2 = x_0 \tag{5.79}$$

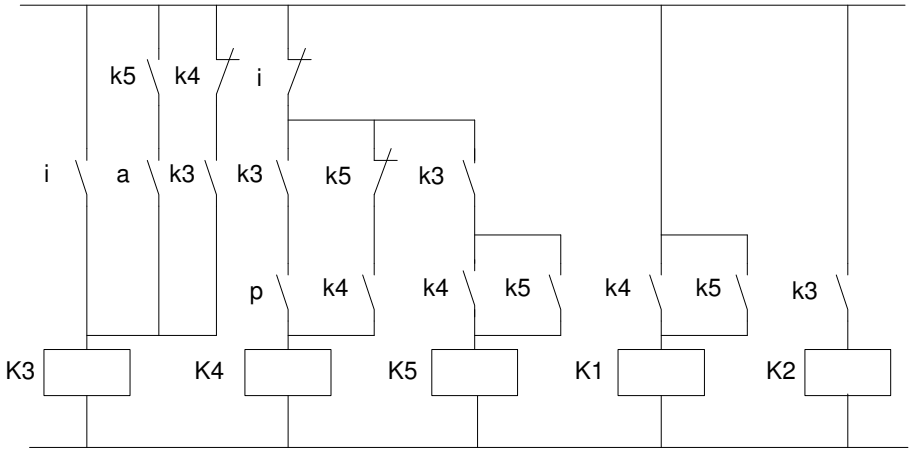


Fig. 5.66 Automatul pentru oprirea unui motor după o rotaţie implementat cu contacte şi relee după transformarea din APL

6. Sisteme de reglare automată

Sistemul de *reglare automată* reprezintă o formă particulară de sistem, studiat cel mai adesea pe baza unei scheme tehnologice sau scheme bloc. Aceste scheme consideră sistemul de reglare automată drept o mulțime de blocuri interconectate[134]. Comportarea fiecărui bloc este determinată de funcția lui de transfer.

Comportarea şi performanța instalațiilor automatizate cu ajutorul *automatelor*[135] este asemănătoare cu cea a instalațiilor neautomatizate cu deosebirea că evoluția lor în timp se realizează fără intervenția omului.

Reglarea automată a proceselor cu ajutorul *regulatoarelor*[136] conduce la o comportare şi performanțe complet diferite de procesele neautomatizate. Sistemele de reglare automată pot devenii frecvent instabile.

Regulatorul, alături de *automat* şi *calculator,* este o componentă centrală a sistemului de conducere automată. Dacă modul în care conduce regulatorul este continuu se obțin cele mai bune rezultate atât în regim de urmărire a referinței cât şi în regim de reglare pentru atenuarea efectelor perturbațiilor. Metodele de proiectare a regulatoarelor cele mai elaborate se bazează pe rezultatele obținute în teoria sistemelor automate continue liniare. Acestea sunt principalele motive pentru care regulatoarele continue sunt echipamente de referință în teoria şi practica sistemelor automate. Alte tipuri de regulatoare, cum ar fi regulatoarele numerice sau regulatoarele cu impulsuri modulate în durată, sunt proiectate adeseori plecând de la echivalentul lor continuu liniar.

Regulatoarele continue moderne conțin adeseori pe lângă compensatorul clasic al erorii şi compensatoare situate pe calea de reacție de la perturbație, pe calea de legătură inversă negativă de la ieşire şi pe calea directă de la referința urmărită.

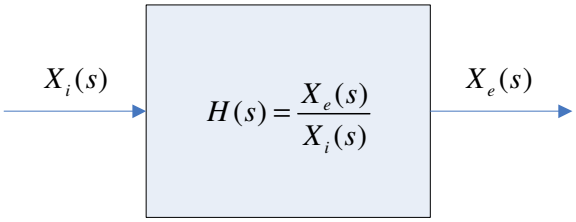


Fig. 6.1 Funcția de transfer a unui bloc definită drept raportul transformatelor Laplace ale semnalelor de intrare şi ieşire cu condiții inițiale nule

[134] Mulțimea relațiilor dintre blocuri formează structura sistemului automat. Sistemele de reglare automată au o structură tipică cu legături inverse negative.

[135] Automate cu evenimente discrete studiate în capitolul precedent

[136] Regulatoare de diferite tipuri: continue, numerice sau directe.

6.1. Sisteme cu conducere manuală şi automată

Un bun punct de plecare pentru prezentarea conducerii automate o prezintă conducerea manuală[137]. O schemă bloc tipică pentru conducerea manuală este prezentată în Fig. 6.2. Semnalele semnificative pentru conducerea procesului din instalaţia G sunt: comanda u(t), execuţia m(t), ieşirea y(t), perturbaţia p(t), zgomotul n(t) şi variaţia parametrilor instalaţiei z(t). Blocul L reprezintă elementul de execuţie, blocurile B şi H reprezintă traductoarele iar blocul A reprezintă procesul prin care perturbaţia p(t) influenţează ieşirea y(t).

In practică, la elaborarea metodelor de conducere, se consideră în majoritatea cazurilor că sistemul condus este *liniar* cu parametrii *invariabili* în timp. Aceasta înseamnă că semnalul z din Fig. 6.2 nu depinde de timp, iar întregul sistem poate fi condus aplicând principiul superpoziţiei care permite considerarea separată a efectelor celor trei intrări u(t), p(t) şi n(t). Efectul total este egal cu suma efectelor parţiale. Chiar din această frază rezultă că semnalul de comandă u(t) trebuie să fie elaborat de operator în aşa fel încât să compenseze efectele perturbaţiei p(t) şi zgomotului n(t). De fapt sarcinile operatorului sunt mult mai complexe. Să enumerăm câteva dintre ele:

- Conducerea de urmărire. Comanda u(t) trebuie să fie astfel elaborată încât ieşirea y(t) să fie identică cu un program de funcţionare în timp determinat de referinţa r(t).
- Conducerea de reglare. Comanda u(t) trebuie să elimine într-o cât mai mare măsură efectul perturbaţiei p(t).
- Conducerea de filtrare. Comanda u(t) trebuie să nu fie influenţată de zgomotul n(t) care apare la ieşirea traductorului H.
- Conducerea de configurarea echipamentului. Operatorul trebuie, de exemplu, să stabilească care sunt pompele de rezervă şi de lucru.
- Conducerea pentru protecţia oamenilor şi a echipamentului. În caz de avarie operatorul trebuie să execute comenzi specifice.
- Conducerea se bazează pe măsurări. Din această cauză operatorul trebuie să aibă grijă să măsoare şi, eventual, să înregistreze variabilele necesare din proces.
- Conducerea în funcţie de apariţia unor evenimente discrete, de exemplu conducerea sistemului de siguranţă la foc în funcţie de semnalele de avarie.

[137] Unele procese nu pot fi conduse manual din cauza constantelor de timp foarte mici sau foarte mari. Algoritmii de conducere automată nu urmăresc, în majoritatea cazurilor, metodele de conducere manuală. De exemplu, componentele integrală şi derivativă a algoritmului PID nu sunt deloc intuitive.

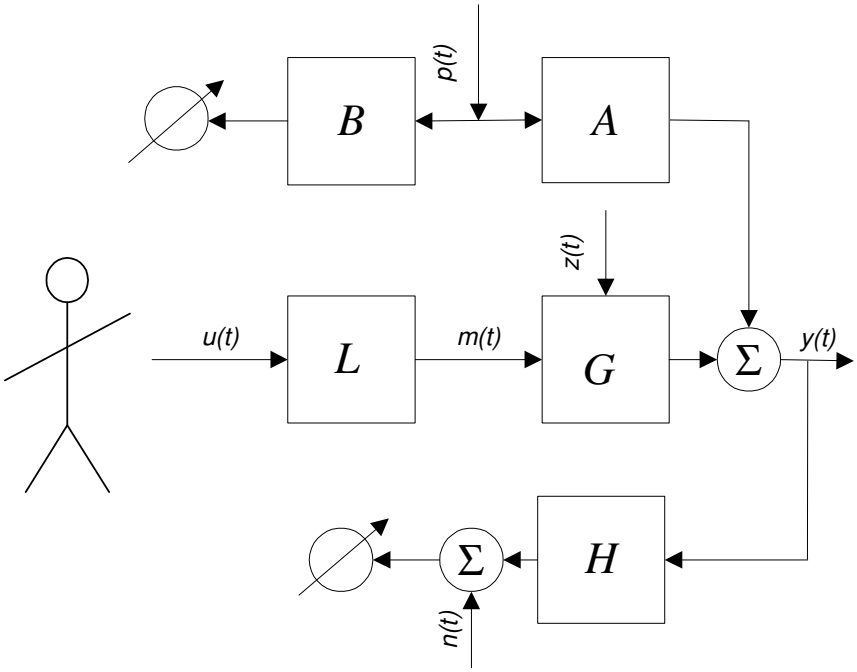


Fig. 6.2 Schema bloc a unui sistem cu conducere manuală a instalației.

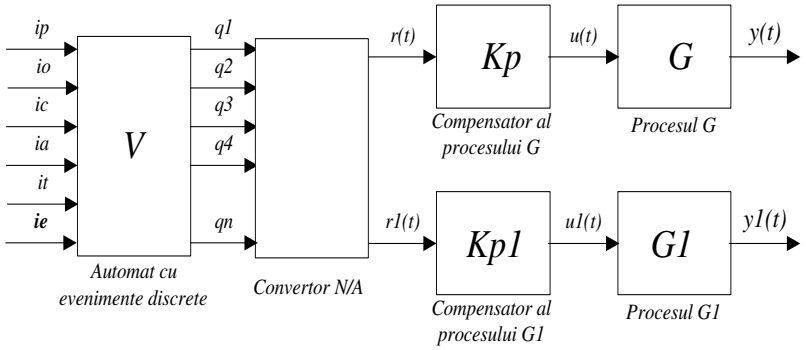


Fig. 6.3 Sistem cu conducerea automată a instalației în regim de urmărire folosind un automat şi compensatoare ale proceselor din instalație

Dacă instalaţia poate fi considerată liniară, sau liniarizabilă pe porţiuni, fiecăreia dintre sarcinile de conducere manuală menţionate îi corespunde un sistem specific de automatizare a instalaţiilor. Să considerăm câteva cazuri.

Conducerea automată în regim de urmărire a proceselor din instalaţie este prezentată în Fig. 6.4. Din punct de vedere al automatizării instalaţiei se remarcă compensatoarele proceselor $K_p(s)$, $K_{p1}(s)$ şi automatul cu evenimente discrete V.

Compensatoarele şi automatul funcţionează cu semnale de conducere diferite. Automatul foloseşte semnale discrete logice iar compensatoarele folosesc semnale continue. Din această cauză legătura dintre ele se realizează printr-un convertor numeric-analogic de semnale. Datorită semnalelor diferite cele două echipamente sunt analizate şi proiectate cu ajutorul unor instrumente teoretice diferite, deşi în practică sistemul de conducere nu se poate dispensa de nici unul.

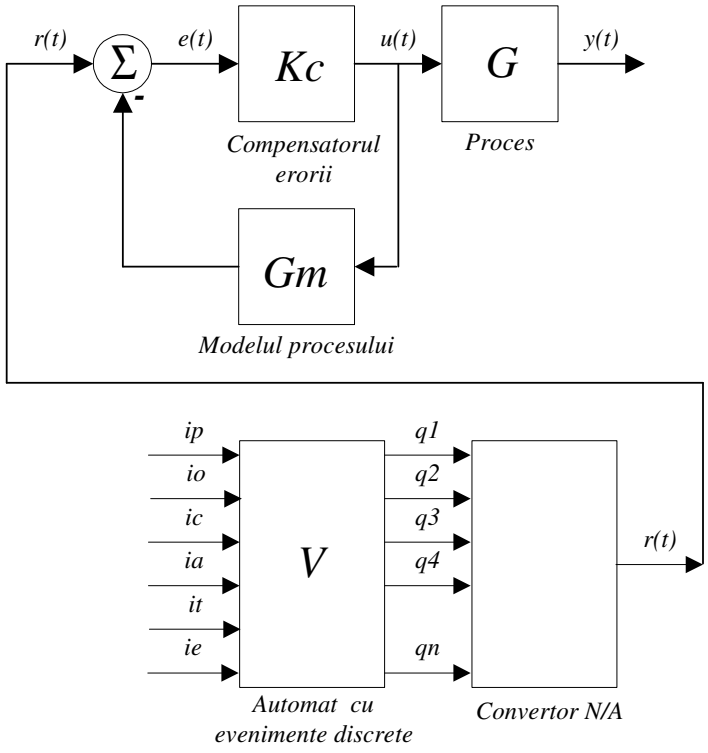


Fig. 6.4 Sistem cu conducere automată a instalaţiei în regim de urmărire folosind un compensator al procesului realizat cu ajutorul reacţiei negative.

Automatul V cu evenimente discrete are la intrare, în cazurile mai simple, următoarele semnale logice exogene:

- ***ip*** evenimente de pornire sau / şi iniţializare. Orice instalaţie de automatizare trebuie că conţină cel puţin un buton de pornire. Conducerea cu automate programabile necesită şi un buton de iniţializare[138]
- ***io*** evenimente de oprire.
- ***ic*** evenimente de configurare. În foarte multe situaţii instalaţia poate avea diferite configuraţii. Înainte de punerea sa în funcţiune , în regim de conducere manuală sau automată, trebuie să se stabilească structura sistemului. De exemplu, care este pompa de lucru şi care pompă funcţionează în regim de rezervă tehnologică.
- ***ia*** evenimente de avarie. Diferite semnale din procesele instalaţiei pot fi indicatori de avarie La apariţia acestor semnale logice instalaţia de conducere automată trebuie să realizeze acţiunile necesare protecţiei oamenilor şi echipamentelor tehnologice. Datorită importanţei deosebite pe care o au aceste acţiuni ele sunt realizate de echipamente specializate care formează instalaţia de protecţie automată.
- ***it*** evenimente de întârziere, de orar şi calendar. Printre altele, aceste evenimente ajută la generarea programului de referinţă r(t) pentru mărimea de ieşire a instalaţiei.
- ***ie*** alte evenimente din proces care ajută la conducerea instalaţiei, de exemplu, prin pornirea şi oprirea unor echipamente.

Ieşirile automatului cu evenimente discrete sunt semnalele logice q1, q2, … qn. Pentru instalaţia de automatizare în regim de urmărire aceste semnale contribuie la generarea referinţelor r(t) şi r1(t) dar ele au un rol important şi pentru echipamentele de automatizare care asigură pornirea, oprirea, configurarea, protecţia şi conducerea instalaţiei în funcţie de evenimente discrete.

În Fig. 6.3 sistemul automat este format din două părţi, una cu semnale discrete logice la care conducerea se face de către automat şi alta cu semnale continue la care conducerea se face de către compensatoare. Sistemele moderne conţin însă şi o a treia parte, cu semnale discrete eşantionate la care conducerea se face de către un microcalculator. Din punct de vedere funcţional la aceste sisteme microcalculatorul îndeplineşte acelaşi rol ca şi automatul sau compensatorul şi din această cauză nu vor fi luate în considerare în cele ce urmează.

Proiectarea şi analiza sistemului automat se face pe baza relaţiilor dintre intrările şi ieşirile elementelor sale. Pentru toate cele trei tipuri de sisteme automate, discret logic, continuu sau discret cu eşantionare, relaţiile de intrare – ieşire pot fi exprimate atât grafic cât şi analitic. In prezent sunt folosite mai mult grafurile[139] pentru sistemele discrete logice şi relaţiile analitice[140] pentru sistemele continue sau discrete cu eşantionare. Evident că pentru proiectarea asistată de calculator şi în cazul sistemelor discrete logice trebuiesc folosite metode analitice matriciale.

---

[138]La automatele cu contacte şi relee iniţializarea se face în mod implicit prin oprire care decuplează alimentarea cu energie electrică.

[139] Reţeaua Petri sau grafcetul.

[140] Ecuaţii matriciale diferenţiale, algebrice de variabilă complexa sau cu diferenţe.

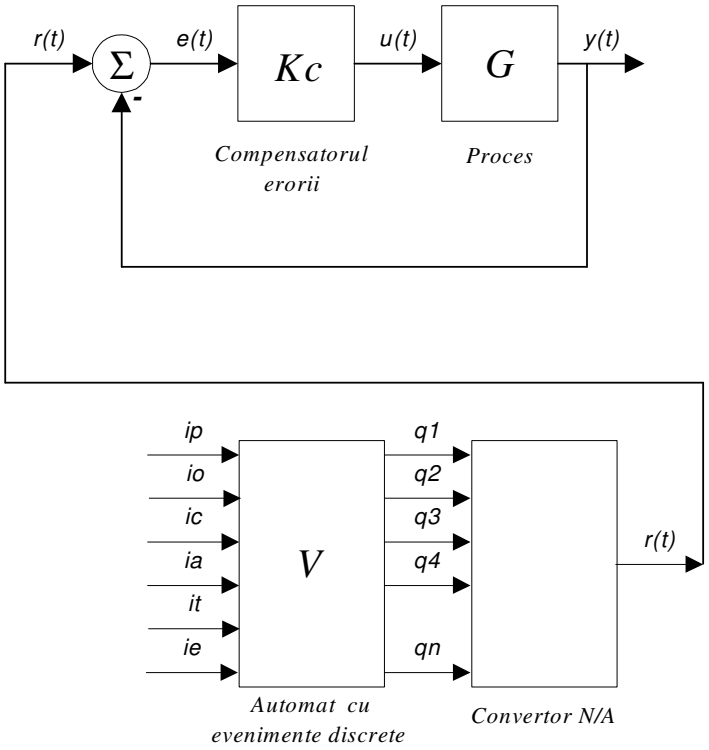


Fig. 6.5 Sistem cu conducere automată a instalației în regim de urmărire pentru situația în care modelul este identic cu procesul.

Relația intrare – ieşire a părții continue, corespunzătoare procesului G(s), a sistemului de conducere automată din Fig. 6.3 este:

$$Y(s) = G(s)K_p(s)R(s) \tag{6.1}$$

în care R(s) şi Y(s) sunt transformatele Laplace ale semnalelor de referință şi de ieşire iar $K_p(s)$ şi G(s) sunt funcțiile de transfer ale compensatorului şi procesului.

În regim de urmărire procesul trebuie să fie astfel condus încât ieşirea procesului să fie identică cu referința. Punând această condiție în relația (6.1) rezultă valoarea compensatorului:

$$K_p(s) = \frac{1}{G(s)} \tag{6.2}$$

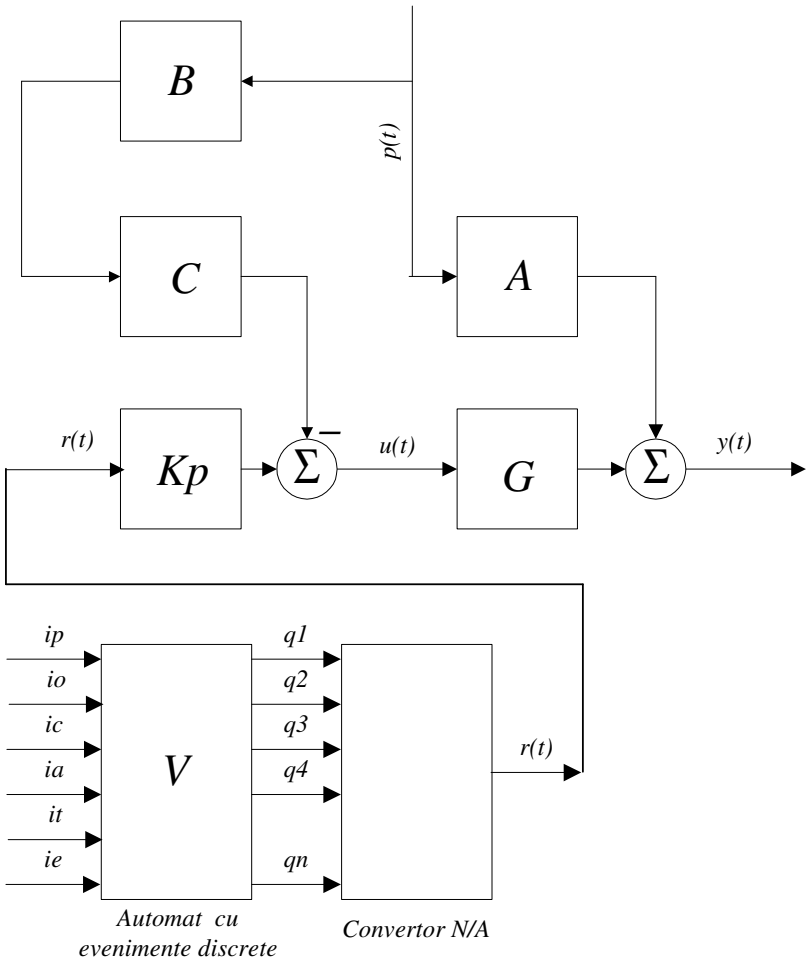


Fig. 6.6 Sistem cu conducerea automată a instalației în regim de reglare folosind reacția de la perturbație[141].

Din păcate compensatorul realizat cu ajutorul acestei relații nu este , în general, realizabil fizic deoarece conține derivatoare sau / şi elemente anticipatoare. Se fac diferite aproximații, de exemplu se consideră relația (6.2) numai în regim staționar. O soluție mai bună este prezentată în Fig. 6.4. Compensatorul procesului $K_p(s)$ este aproximat cu ajutorul unei bucle cu reacție negativă care cuprinde un model $G_m(s)$ al procesului şi un compensator proporțional al erorii K_c. Funcția de transfer a compensatorului procesului $K_p(s)$ va fi:

[141] Feedforward.

$$K_p(s) = \frac{K_c}{1 + K_c G_m(s)} \tag{6.3}$$

Dacă

$$K_c \to \infty \tag{6.4}$$

atunci relația (6.2) este îndeplinită. Din păcate condiția (6.4) conduce la instabilitatea sistemului automat şi atunci se folosesc valori finite pentru constanta de proporționalitate K_c astfel încât compensarea dinamică a procesului în regim de urmărire este doar aproximativă.

Dacă modelul procesului $G_m(s)$ este identic cu procesul G(s)

$$G_m(s) = G(s) \tag{6.5}$$

atunci în locul sistemului din Fig. 6.4 se poate folosi sistemul automat din Fig. 6.5 şi se obține o buclă standard în regim de urmărire cu un generator de referință format dintr-un automat care asigură şi alte funcții, dintre cele prezentate, ale conducerii automate.

Relația dintre ieşirea şi referința sistemului automat este:

$$Y(s) = \frac{K_c G(s)}{1 + K_c G(s)} = T(s)R(s) \tag{6.6}$$

în care

$$T(s) = \frac{K_c G(s)}{1 + K_c G(s)} \tag{6.7}$$

este funcția de sensibilitate complementară a buclei

Un sistem de conducere automată a proceselor termice din instalațiile pentru clădiri şi locuințe funcționează atât în regim de urmărire cât şi în regim de reglare şi este prezentat în *Fig. 6.6*. Înlăturarea perturbației se face prin reacție negativă de la perturbație. Aceasta presupune că perturbația poate fi măsurată. Ieşirea sistemului este:

$$Y(s) = G(s)K_p(s)R(s) + \left[A(s) - G(s)C(s)B(s)\right]P(s) \tag{6.8}$$

Pentru înlăturarea efectului perturbație compensatorul C(s) trebuie să aibă valoarea:

$$C(s) = \frac{A(s)}{B(s)G(s)} \tag{6.9}$$

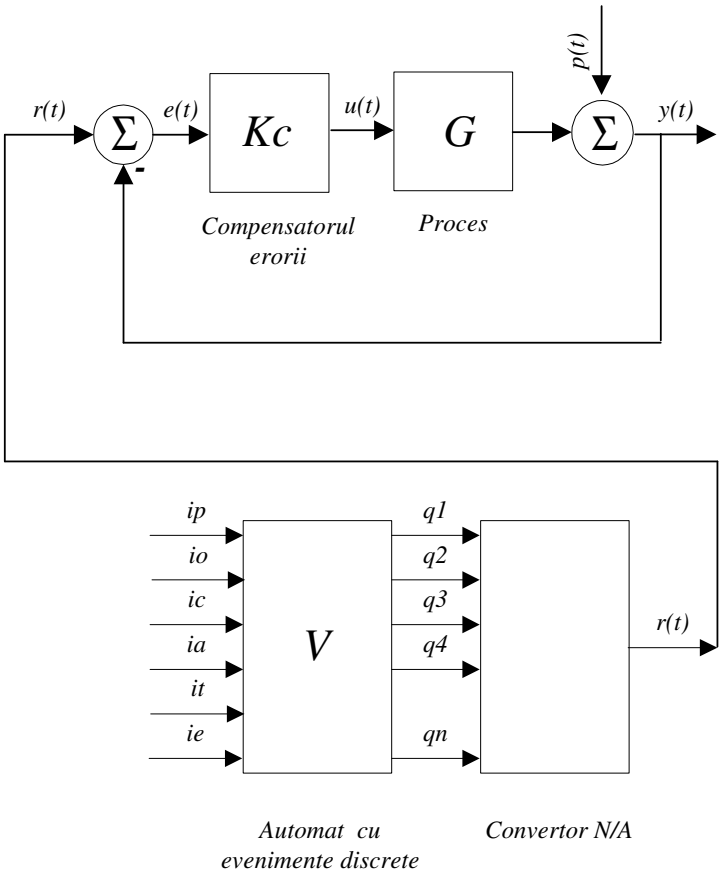


Fig. 6.7 Sistem cu conducerea automată a instalației în regim de reglare cu reacție[142] negativă.

Din păcate funcția de transfer a perturbației A(s) nu este cunoscută şi atunci efectul compensării este doar parțial. O variantă mai eficientă de conducere automată care înlătură şi efectul perturbației este prezentată în Fig. 6.7. În regim de urmărire R(s)=0 şi ieşirea sistemului automat este:

$$Y(s) = \frac{1}{1 + K_c(s)G(s)} P(s) = S(s)P(s) \qquad (6.10)$$

în care

$$S(s) = \frac{1}{1 + K_c G(s)} \qquad (6.11)$$

[142] Feedback, legătură inversă.

este funcția de sensibilitate a buclei din Fig. 6.7.

Intre funcțiile de sensibilitate S(s) şi sensibilitate complementară T(s) există relația

$$S(s) + T(s) = 1 \qquad (6.12)$$

Eliminarea efectului perturbației se face dacă este îndeplinită condiția (6.4) valabilă în regimul de urmărire. Ca şi în cazul sistemului automat din Fig. 6.5 constanta de proporționalitate K_c nu poate fi prea mare deoarece sistemul automat tinde să devină instabil.

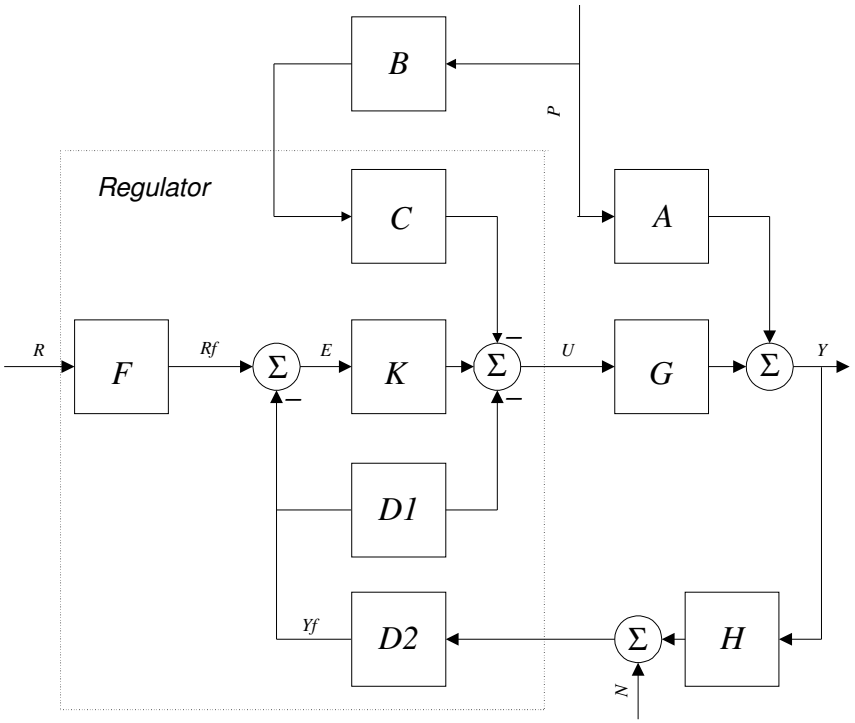


Fig. 6.8 Sistem cu conducerea automată continuă a instalației cu regulator cu două grade de libertate.

In mod frecvent se folosesc în practică diferite combinații ale structurilor prezentate până acum. În Fig. 6.8 se prezintă o situație generală în care se foloseşte atât feedback-ul cât şi feedforward-ul. Sistemul automat continuu funcționează atât în regim de urmărire cât şi în regim de reglare şi regim de filtrare. Pentru simplificarea figurii automatul cu evenimente discrete nu mai este prezentat, dar el este indispensabil pentru pornirea, oprirea, protecția, generarea referinței şi alte funcțiuni curente ale instalației de automatizare.

Îmbunătățirea performanțelor sistemului automat în regim de urmărire şi în regim de stabilizare se face adoptând pentru compensatorul erorii K(s) din Fig. 6.8 funcții de transfer diferite de cea proporțională K_c folosită până acum. Eroarea

indicială staţionară devine nulă dacă compensatorul are , în afară de componenta proporţională şi o componentă integrală. Eroarea staţionară pentru semnale de intrare periodice poate fi micşorată dacă se foloseşte şi o componentă derivativă. În felul acesta se ajunge la regulatorul clasic proporţional – integral – derivativ PID folosit frecvent în practică.

Diferitele performanţe impuse sistemului automat în cele trei regimuri de funcţionare, urmărire, reglare şi filtrare conduc la cerinţe contradictorii pentru funcţia de transfer a compensatorului erorii datorită condiţii (6.12). O rezolvare parţială a problemei se realizează prin introducerea unui bloc nou, de prefiltrare F, în schema regulatorului din Fig. 6.7. Metoda de conducere automată a procesului G(s) din instalaţie foloseşte în acest caz un regulator cu două grade de libertate.

Automatele cu evenimente discrete şi regulatoarele cu compensatoare a erorii sunt echipamente de bază în cadrul metodelor generale de conducere automată. În cazul conducerii automate a serviciilor oferite de clădiri şi locuinţe se folosesc însă frecvent şi metode de conducere prin compensarea procesului şi compensarea perturbaţiei. Câteva posibilităţi de integrare a tuturor acestor metode au fost prezentate în această lucrare.

6.2. Regulatoare PID continue

Echipamentele moderne de conducere automată oferă posibilitatea folosirii unei game foarte largi de algoritmi. Unul dintre algoritmii cei mai frecvent folosiţi este algoritmul proporţional integral derivativ PID, din care s-au identificat peste 297 de variante utilizate în regulatoarele comerciale. De exemplu, companiile National Instruments, ABB , Bailey, Fisher, Foxboro, Honeywell, Moore Products, Yokogawa şi altele, comercializează regulatoare pentru care denumirea algoritmului, terminologia întrebuinţată pentru descrierea lui şi a unităţilor de măsură este diferită.

Schema bloc a unui sistem de reglare automată clasic este prezentată în Fig. 6.9. Cele trei intrări ale sistemului automat sunt referinţa R, perturbaţia P şi zgomotul N de la ieşirea traductorului. Regulatorul este format dintr-un comparator şi compensatorul K. Blocul G de pe calea directă modelează elementul de execuţie şi procesul automatizat. Traductorul este reprezentat de blocul H. Funcţia de transfer a unui compensator K de tip PID are următoarea formă generală:

$$K(s) = \frac{U(s)}{E(s)} = K_c\left(1 + \frac{1}{T_i s} + T_d s + q\frac{T_d}{T_i}\right) \qquad (6.13)$$

în care K_c = (100%) / BP este constanta de proporţionalitate a compensatorului , iar BP este banda de proporţionalitate măsurată în procente,

T_i: constanta de timp integral sau timpul de repetare [s / repetare],

T_d: constanta de timp derivativ [s],

q: factor de influenţă.

Algoritmul PID cu factor de influenţă zero q=0, numit *ideal*, este cel preferat de teoreticieni şi prezentat cu precădere în toate manualele şi monografiile consacrate sistemelor automate

Forma algoritmului, numită algoritmul *PID paralel*, este preferată de unele firme şi de unele manuale universitare. În acest caz se pune clar în evidenţă acţiunea proporţională (amplificarea) K_p, acţiunea integrală (restabilirea automată) K_i/s şi acţiunea derivativă $K_d s$.

$$\begin{aligned} &PID\ ideal: && K(s) = K_c\left(1+\frac{1}{T_i s}+T_d s\right) \\ &PID\ paralel: && K(s) = K_p+\frac{K_i}{s}+K_d s \\ &PID\ serie: && K(s) = K_c(1+\frac{1}{T_i s})(T_d s+1) \end{aligned} \qquad (6.14)$$

Algoritmul *PID serie (interactiv)* corespunde valorii q=1 a factorului de influenţă şi se obţine după câteva transformări simple din relaţia generală.

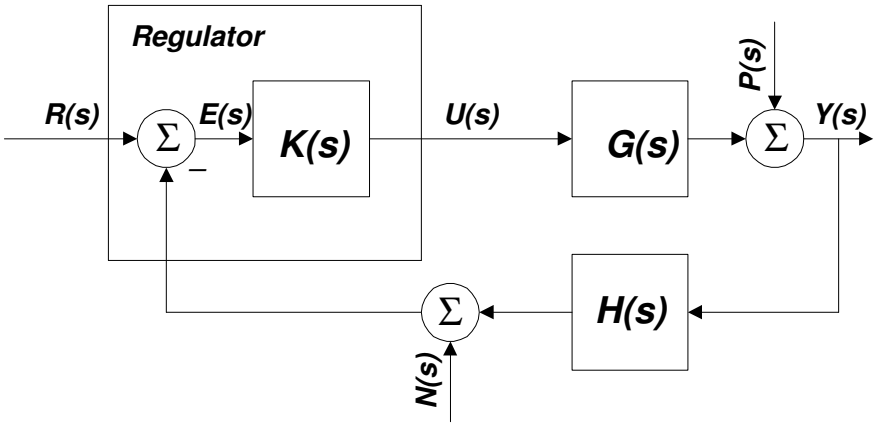


Fig. 6.9 Bucla de reglare standard

Toate cele trei forme ale algoritmului PID sunt folosite în prezent de către producătorii de regulatoare automate. De exemplu, AEG Modicon şi Texas Instruments folosesc tipul ideal, Foxboro şi Fisher au adoptat algoritmul serie, Honeywel are regulatoare PID atât serie cât şi ideale iar Bailey şi Allen Bradley au regulatoare cu algoritmi tip PID ideal şi paralel.

Dacă nu ne interesează firma producătoare şi analizăm regulatoarele automate din alte puncte de vedere, se poate constata că aproape toate regulatoarele analogice electronice şi pneumatice sunt de tip serie. Regulatoarele numerice sunt în cea mai mare parte de tip ideal. Un număr mai mic de regulatoare numerice sunt de tip serie pentru a fi echivalente cu regulatoarele analogice.

Compensatoarele PID se transformă uşor în compensatoare P dacă $T_d=0$ şi $T_i=\infty$, în compensatoare PI dacă $T_d=0$, sau compensatoare PD dacă $T_i=\infty$.

Compensatoarele ideale PID diferă de cele serie numai în cazul în care toate cele trei acțiuni P, I şi D sunt prezente.

Răspunsul compensatorului PI la o eroare treaptă unitară este $u=K_c(1+t/T_i)$. Atunci când $t=T_i$ efectul proporțional al algoritmului se *repetă* (dublează). Din această cauză T_i se măsoară în secunde / repetare. O eroare rampă provoacă răspunsul $u=K_c(t+T_d)$ al compensatorului PD. Dacă $t=T_d$ efectul proporțional al algoritmului se dublează şi în felul acesta poate fi determinată constanta de timp derivativ.

6.2.1. Compensatorul PID analogic.

În documentația tehnică sau în manuale algoritmul PID este prezentat, de obicei, într-una din formele ideală, paralelă sau serie. Algoritmul real, folosit de regulator la conducerea procesului, este însă diferit deoarece termenul T_ds, corespunzător acțiunii derivative, care apare în funcția de transfer a compensatorului PID nu este realizabil fizic. Regulatoarele comerciale analogice folosesc, din acest motiv, aproximarea:

$$T_d s \cong \frac{T_d s}{\alpha T_d s + 1} \tag{6.15}$$

în care α este o caracteristică constructivă a compensatorului, care nu poate fi modificată de către utilizator, cu o valoare fixată undeva între 1/6 şi 1/20.

Cu această aproximație algoritmii PID pentru compensatoarele analogice sunt prezentați adeseori sub această formă:

$$\begin{aligned} &PID\ ideal\ analogic: \quad K(s) = K_c(1+\frac{1}{T_i s}+\frac{T_d s}{\alpha T_d s+1}) \\ &PID\ serie\ analogic: \quad K(s) = K_c(1+\frac{1}{T_i s})(\frac{T_d s+1}{\alpha T_d s+1}) \end{aligned} \tag{6.16}$$

6.2.2. Regulatorul PID cu două grade de libertate.

Regulatoarele moderne, în special cele numerice, au schema bloc prezentată în *Fig. 6.10*. Față de schema clasică din Fig. 6.9 regulatorul are două blocuri noi: prefiltrul F şi blocul D care include acțiunea derivativă şi filtrul zgomotului N.

Funcția de transfer a prefiltrului este, de regulă, de forma

$$F(s) = \frac{\beta T_i s + 1}{T_i s + 1} \tag{6.17}$$

sau echivalentul său numeric obținut prin aproximarea lui s. T_i este constanta de timp integral iar constanta β, cu valori între 0 şi 1, se determină astfel încât răspunsul indicial al sistemului automat în regim de urmărire să aibă

suprareglări mici, $\sigma \leq 7\%$. Existența prefiltrului permite acordarea compensatorului PID numai pentru funcționarea în regim de reglare (stabilizare).

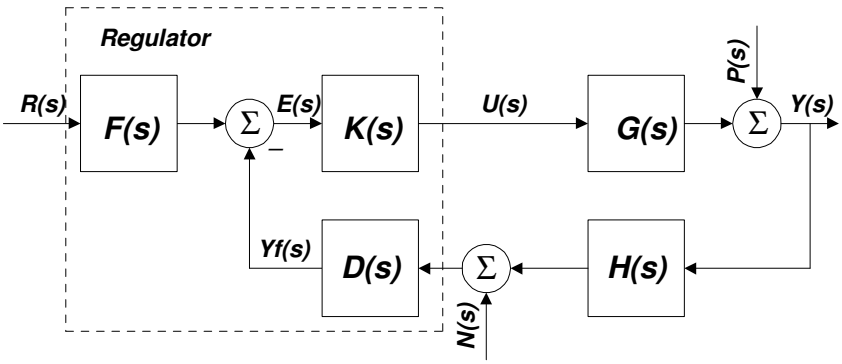


Fig. 6.10 Regulatorul PID cu două grade de libertate

Zgomotul N, atunci când depăşeşte 3% din domeniul de lucru, poate fi filtrat de blocul D cu funcția de transfer, având de regulă, forma

$$D(s) = \frac{T_d s + 1}{\alpha T_e s + 1} \qquad (6.18)$$

sau cu echivalentul său numeric obținut prin aproximarea lui s. În expresia funcției de transfer T_d este constanta de timp derivativ, T_e este perioada de eşantionare şi α determinat astfel încât să filtreze zgomotul fără a reduce performanțele sistemului automat. O valoare aproximativă pentru α, atunci când T_e este aleasă în mod corect, este mai mare decât unu.

Regulatorul PID cu două grade de libertate din *Fig. 6.10* are compensatorul K de tipul proporțional integral. Acțiunea derivativă, atunci când există, este realizată de blocul D care face în acelaşi timp şi o operație de filtrare a zgomotului N.

O altă variantă de regulator PID cu două grade de libertate este prezentată în Fig. 6.11. În acest caz acțiunea derivativă este dată de funcția de transfer $D1=T_d s$ iar filtrarea zgomotului este realizată de funcția de transfer $D2(s)=1/(\alpha T_e s+1)$. Variantele numerice se obțin prin aproximarea lui s.

Acordarea regulatorului PID cu două grade de libertate se face prin stabilirea constantelor K_c, T_i, T_d, α şi β. Dacă α=0 şi β=1 se obține un regulator cu un singur grad de libertate, cu algoritmul PID ideal şi cu acțiunea derivativă scoasă de pe calea directă.

6.2.3. Regulatorul PI cu impulsuri modulate în durată

Aproape toate regulatoarele moderne PID au şi o variantă de algoritm PID cu impulsuri modulate în durată. Cel mai frecvent este folosit regulatorul PI cu impulsuri modulate în durată prezentat în Fig. 6.12. Compensatorul acestui regulator este de tip PD şi are la ieşire un tren de impulsuri cu durata variabilă. Dacă aceste impulsuri sunt integrate de un servomotor se obține un efect aproximativ de tip proporțional integral PI.

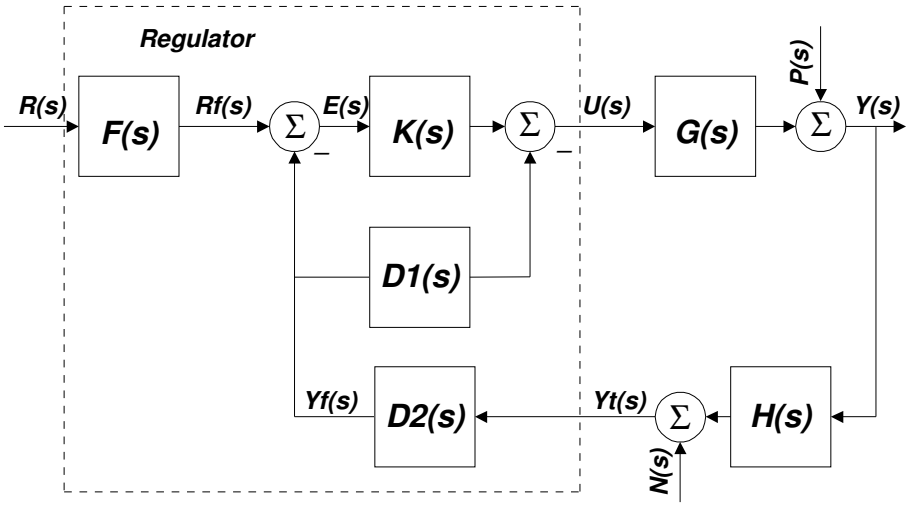


Fig. 6.11 Buclă de reglare cu compensator pe calea de reacţie

Compensatorul PD cu IMD este format dintr-un sistem cu reacţie negativă care are pe cale directă un bloc tripoziţional iar pe calea inversă un bloc cu o funcţie de transfer de ordinul unu. Impulsurile care apar la ieşirea acestui compensator au amplitudinea de plus i_1 sau minus i_1. Dacă eroarea nu este prea mică sau prea mare impulsurile au o lăţime proporţionala cu eroarea şi o perioada de apariţie constantă. Pentru acest domeniu al erorilor compensatorul PD împreună cu servomotorul realizează un algoritm aproximativ PI cu IMP.

$$PI\ cu\ IMD:\quad daca\ \frac{\Delta e}{i_r} = 0{,}3\ldots 0{,}7\ atunci:$$

$$K(s) = \frac{U(s)}{E(s)} \cong \frac{u_M}{i_r}\frac{T}{T_M}\left(1+\frac{1}{Ts}\right)\frac{1}{T_1 s+1} \qquad (6.19)$$

$in\ care:$

$$i_r = Ki_1 \quad , \quad T_1 = T\frac{\Delta e}{i_r} \quad , \quad \Delta e = e - z$$

Se observă că parametrii regulatorului PI cu IMD depind de parametrii servomotorului, de parametrii blocului de reacţie şi de mărimea erorii.

Pentru erori mici regulatorul PI cu IMD funcţionează aproximativ la fel cu un regulator tripoziţional. Parametrii acestuia, zona de insensibilitate 2z şi lăţimea ciclului histerezis 2h, trebuiesc luaţi în considerare de utilizator.

Regulatoarele PI cu IMD sunt preferate adeseori deoarece folosesc motoare asincrone cu rotorul în scurtcircuit comandate de impulsurile de la ieşirea regulatorului prin intermediul unor contactoare.

Numărul mare de variante de algoritmi PID al regulatoarelor moderne provoacă dificultăţi nu numai la înţelegerea modului de funcţionare şi definirea parametrilor caracteristici ci şi la acordarea lor folosind criteriile empirice existente.

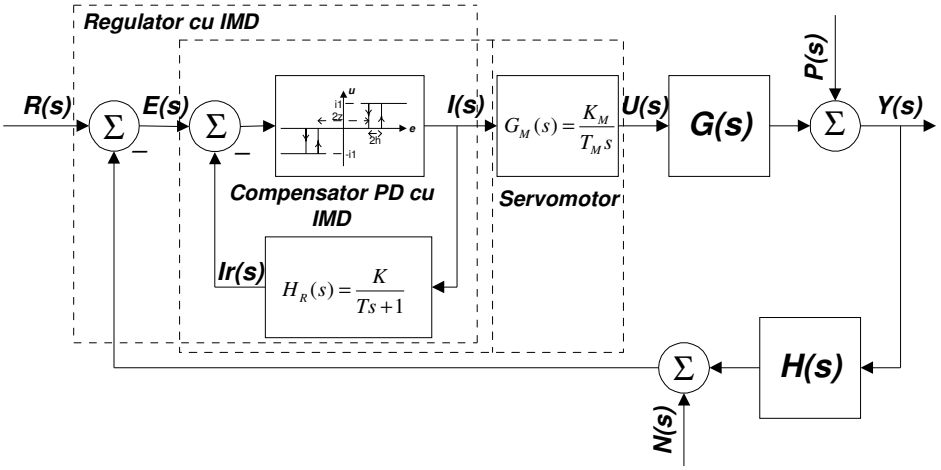


Fig. 6.12 Sistem automat cu regulator cu impulsuri modulate în durată

6.3. Proiectarea sistemelor automate liniare

Proiectarea sistemelor automate necesită în practică un efort ciclic în care se iterează prin mai multe etape de modelare, proiectare, simulare, testare şi implementare. Proiectarea depinde de mai mulţi factori dintre care unul important este determinat de faptul că sistemul automat face parte dintr-un echipament produs în scopuri comerciale. În această situaţie necesitatea unui preţ scăzut şi posibilităţile oferite de echipamentele existente pe piaţă influenţează hotărâtor soluţia aleasă.

Inginerul automatist proiectant se poate găsi în una dintre următoarele situaţii:

- Inginerul automatist participă la dezvoltarea sistemului încă de la început. De multe ori în aceste situaţii se impun performanţe numai în regim staţionar de funcţionare. Nu este deci nici o mirare că dificultăţile apar mai târziu în exploatarea sistemului.
- Inginerul automatist acordează sistemul de conducere în funcţiune pentru a realiza performanţele prevăzute cât mai bine posibil. Această fază necesită o înţelegere profundă a modului de funcţionare a sistemelor cu reacţie negativă şi a posibilităţilor de intervenţie.
- Inginerul automatist întreţine şi îmbunătăţeşte funcţionarea instalaţiei. Necesitatea îmbunătăţirii provine din mai multe direcţii. Pot apare , de exemplu, echipamente noi cu posibilităţi mai bune de conducere sau se modifică legislaţia de protecţie a mediului înconjurător.

- Inginerul automatist realizează şi studii de prognoză asupra perspectivelor de dezvoltare a instalaţiilor conduse.

Succesul în conducerea automată este condiţionat de o abordare sistemică. Aceasta include cunoaşterea tuturor elementelor, a relaţiilor dintre ele şi a scopurilor sistemului. Într-o sarcină tipică de proiectare acestea sunt:

- Instalaţia condusă. De fapt este vorba de procesele din instalaţie care constituie obiectul conducerii cu ajutorul automatelor şi/sau regulatoarelor. Trebuiesc cunoscute bilanţurile energetice şi fluxurile de masă din sistem. Dimensiunile fizice ale echipamentului şi modul în care acestea influenţează performanţele sistemului sunt de asemenea importante şi trebuiesc înţelese. Toate aceste cunoştinţe trebuiesc incluse într-un *model* al procesului condus şi care va fi rafinat pe parcursul fazelor de proiectare.
- Scopurile conducerii. Ce trebuie să se obţină? Reducere energiei consumate, un confort sporit, etc. Care sunt variabilele care trebuiesc controlate pentru atingerea acestor obiective. Care este nivelul de performanţă necesar.
- Traductoarele. Conducerea unui sistem nu poate fi mai bună decât traductoarele folosite. Tot timpul apar noi realizări în domeniul acesta. De exemplu sunt folosite în prezent traductoarele inteligente sau traductoarele virtuale care obţin informaţiile necesare din diferite observaţii.
- Elementele de execuţie. Echipamente nepotrivite sau de slabă calitate folosite la comanda instalaţiilor sunt adeseori sursa dificultăţilor apărute pe parcursul exploatării.
- Comunicaţiile dintre componente. Interconectarea traductoarelor, elementelor de execuţie, automatelor şi regulatoarelor se face prin intermediul sistemului de comunicaţie. O clădire mare sau o uzină necesită ca mii de semnale să fie transmise la distanţe relativ mari, până la 1000 m. Transmiterea datelor în timp real necesită controlul riguros al întârzierilor spre deosebire de transmiterea semnalelor vocale. Toate sistemele de conducere automată depind de cunoaşterea precisă nu numai a *ce* se întâmplă dar şi de *când* se întâmplă. De exemplu, există tendinţa în creştere de a folosi conexiunile tip Ethernet pentru transmisiile de date în conducerea automată. Totuşi acestea nu sunt potrivite în acest scop deoarece modul lor de funcţionare presupune că atunci când apare imposibilitatea conectării se încearcă reluarea transmisiei ceva mai târziu cu o întârziere aleatoare.
- Algoritmii de conducere şi calculele necesare implementării lor. Metoda de conducere este influenţată decisiv de metodele de calcul numeric sau analogic folosite. Dacă alegerea traductoarelor, elementelor de execuţie şi sistemului de transmisie este influenţată puternic de consideraţii tehnologice şi economice, algoritmul şi implementarea lui oferă mult mai multe grade de libertate în efortul de a asigura performanţele dorite ale sistemului de conducere. Din păcate algoritmi necorespunzători pot conduce nu numai la micşorarea performanţelor dar şi la micşorarea robusteţii sau chiar la instabilitatea sistemului automat.
- Arhitectura sistemului şi dispozitivele de interfaţare. La o primă vedere se poate aprecia că conducerea centralizată la care toate traductoarele şi elementele de execuţie sunt conectate la un singur echipament de conducere este cea mai bună soluţie deoarece se bazează pe toate informaţiile disponibile.

Totuşi aceasta este rareori cea mai bună soluţie în practică datorită problemelor provocate de complexitate, preţ de cost, întârzierilor mari datorate comunicaţiei şi timpilor de calcul, fiabilităţii şi întreţinerii. Uneori aceste probleme pot fi evitate printr-o arhitectură corespunzătoare. Un exemplu în acest sens îl constituie folosirea structurii tip feed – forward la conducerea automată a instalaţiilor din clădiri şi locuinţe.

- Perturbaţiile posibile. Unul dintre scopurile principale al automatizării îl constituie micşorarea influenţei perturbaţiilor de diferite tipuri, cum ar fi de exemplu variaţiile de sarcină ale instalaţiilor. Din păcate perturbaţiile au un caracter aleator pronunţat şi caracteristicile lor sunt puţin cunoscute. Este cu totul remarcabil că în aceste condiţii sistemele automate bine proiectate reuşesc să atenueze puternic efectul lor.
- Incertitudinea modelelor folosite. Conducerea automată poate fi mai bună dacă se cunoaşte mai bine sistemul condus. Aceasta nu este cazul în practică şi faptul că totuşi automatizarea funcţionează bine în multe cazuri este un fapt cu totul remarcabil. Metodele moderne de proiectare pun un accent tot mai mare pe robustete sistemului automat, adică pe caracteristica sa de a fi puţin sensibil la schimbări ale procesului condus.

6.3.1. Analiza sistemelor automate

Printre datele achiziţionate în clădiri şi locuinţe au o importanţă deosebită cele referitoare la procesele fizice din instalaţii numite semnale. Prelucrarea semnalelor (Signal Processing) pentru stabilirea anumitor performanţe sau prognoza unor evenimente presupune reprezentarea lor în domeniul timp sau frecvenţă. Un semnal periodic poate fi reprezentat ca o sumă (serie Fourier) de componente sinusoidale. Un grafic care reprezintă valoarea amplitudinii sinusoidelor în funcţie de frecvenţă. sau pulsaţie, reprezintă o prezentare a semnalului în domeniul frecvenţă.

Pentru toate sistemele liniare, inclusiv cele automate, se aplică principiul superpoziţiei conform căruia efectul unei sume de cauze aditive este egal cu suma efectelor provocate de fiecare cauză în parte. Din această cauză comportarea sistemelor ca răspuns la semnale de intrare complexe poate fi studiată ca o sumă a comportărilor pentru diferite semnale sinusoidale de intrare. Acesta este studiul sistemelor în domeniul frecvenţă.

În continuare se prezintă câteva caracteristici ale sistemelor automate determinate în domeniul timp, *Tab. 6.1* şi în domeniul frecvenţă, *Tab. 6.2*.

Analiza sistemelor automate în domeniul timp şi analiza în domeniul frecvenţă sunt în mare echivalente. Aceste analize prezintă unele particularităţi, avantaje şi dezavantaje care sunt menţionate în continuare.

- ✓ Analiza în domeniul timp este avantajoasă, în cazul sistemelor automate, prin metode experimentale.
- ✓ Analiza în domeniul frecvenţă este avantajoasă prin metode manuale, folosind calculul simbolic.
- ✓ Ambele tipuri de metode, în domeniul timp şi în domeniul frecvenţă, pot beneficia din plin de ajutorul dat de un calculator.
- ✓ Toate sistemele fizice prezintă, în ultima instanţă, anumite neliniarităţi. Analiza sistemelor neliniare este mai uşoară în domeniul timp.

- ✓ Modelele liniare ale sistemelor fizice sunt mult mai simple şi există o teorie bine pusă la punct pentru ele atât în domeniul timp cât şi în domeniul frecvență. Proiectarea asistată de calculator este mult mai bine pusă la punct pentru aceste modele.
- ✓ Metodele de aproximare ale sistemelor fizice prin modele liniare include: a) *transformări,* cum ar fi schimbările de variabilă, b) *aproximări,* de exemplu dezvoltări în serie Taylor în vecinătatea unui punct, identificarea proceselor cu ajutorul cutiilor negre, etc.
- ✓ Câteva modele liniare sunt: modelul tip proporțional, modelul de ordinul unu, modelul de ordinul doi, integratorul şi modelul cu întârziere pură împreună cu aproximarea sa sub formă de fracție rațională. Aceste modele sunt cel mai des observate în practică iar modele liniare mai complexe pot fi descompuse într-o sumă de aceste modele prin metoda descompunerii în fracții parțiale.
- ✓ Calculul funcției de transfer pentru o frecvență oarecare conduce la un rezultat sub forma unui număr complex.
- ✓ Sistemele automate continue pot fi: a) sisteme stabile dacă toți polii sunt situați în semiplanul stâng, b) sisteme la marginea de stabilitate dacă cel puțin un pol se găseşte pe axa imaginară şi nici un pol nu se află în semiplanul drept, c) sisteme instabile dacă cel puțin un pol se găseşte în semiplanul drept, şi d) sisteme de fază neminimă dacă cel puțin un zero se găseşte în semiplanul drept.
- ✓ Răspunsul unui sistem liniar la o intrare arbitrară poate fi descompus în două părți: a) răspunsul natural, care este funcție de condițiile inițiale dar este independent de intrare; dacă sistemul liniar este stabil răspunsul natural se amortizează şi tinde către zero, b) răspunsul forțat care este funcție numai de intrare şi independent de condițiile inițiale.
- ✓ Modelele sistemelor liniare sub formă de funcție de transfer pot fi obținute prin aceste metode: a) experimental, b)aplicând transformarea Laplace modelului sub formă de ecuație diferențială şi c) aplicând transformarea Laplace răspunsului sistemului la un impuls Dirac.
- ✓ Teoretic răspunsul unui sistem liniar modelat cu ajutorul funcției de transfer se obține calculând ieşirea şi aplicând transformarea Laplace inversă asupra ei. Totuşi, în practică se preferă adeseori să se transforme funcția de transfer în domeniul timp şi să se rezolve ecuația diferențială prin metode numerice.
- ✓ Toate modelele conțin erori de modelare. Se presupune, în general, că erorile de modelare cresc cu frecvența.

Tab. 6.1

Caracteristica	***Semnificaţia***
Constanta de proporţionalitate	Determină cum sistemul amplifică sau atenuează semnalul de intrare în regim staţionar
Timpul de creştere	Determină cât de rapid reacţionează sistemul la variaţia intrării
Fracţiunea de amortizare critică	Determină cât de rapid se amortizează oscilaţiile răspunsului
Durata procesului tranzitoriu	Precizează începutul regimului staţionar
Suprareglarea	Determină cât de mult se abate răspunsul de la valoarea staţionară

Tab. 6.2

Caracteristica	***Semnificaţia***
Diagramele Bode şi hodograful Nyquist	Permit determinarea marginilor de fază, de amplitudine şi de modul şi precizează comportarea sistemului pentru diferite intrări sinusoidale.
Polii	Rădăcinile numitorului funcţiei de transfer a sistemului. Determină stabilitatea şi împreună cu zerourile stabilesc caracteristicile în regim tranzitoriu.
Zerourile	Rădăcinile numărătorului funcţiei de transfer a sistemului. Nu determină stabilitatea sistemului dar stabilesc suprareglarea şi împreună cu poli au un impact profund asupra caracteristicilor sistemului în regim tranzitoriu.
Locul rădăcinilor	Locul geometric al polilor şi zerourilor sistemului automat la modificarea constantei de proporţionalitate a buclei deschise
Adecvarea strictă	Sistemul are mai mulţi poli decât zerouri şi este cauzal deci poate fi implementat.
Banda de trecere	Domeniul de frecvenţă în care sistemul influenţează foarte puţin amplitudinea sinusoidelor de intrare

6.3.2. Schema bloc a sistemului automat

Schema bloc a unui sistem automat modern folosit în instalaţiile pentru construcţii este prezentată în Fig. 6.8. Dacă considerăm pentru simplificare că D1=0 şi D2=1 atunci relaţia dintre ieşirea sistemului Y(s) şi cele trei intrări ale sale, referinţa R(s), perturbaţia P(s) şi zgomotul N(s), este următoarea:

$$Y = \frac{KG}{1+KGH} FR + \frac{A-BCG}{1+KGH} P - \frac{KG}{1+KGH} N \qquad (6.20)$$

Se observă că dacă compensatorul perturbației are funcția de transfer C=A/BG atunci influența perturbației este nulă. Din păcate funcțiile de transfer G(s) şi în special A(s) nu sunt cunoscute şi compensatorul C(s) nu poate fi determinat cu precizie. Tot din relația (6.20) se constată însă, că dacă compensatorul erorii există şi are o funcție de transfer K(s) destul de mare, efectul perturbației este dramatic scăzut chiar în prezența unui risc de necunoaştere a proceselor conduse. Din această cauză vom considera în continuare numai efectul reacției negative în eliminarea perturbației P(s) pe baza schemei bloc din Fig. 6.9.

Funcția de transfer $H_u(s)$ a sistemului automat din Fig. 6.9 în regim de urmărire este dată de relația (6.21) dacă se consideră traductorul ideal H=1.

$$H_u(s) = \frac{Y(s)}{R(s)} = \frac{K(s)G(s)}{1+K(s)G(s)} \qquad (6.21)$$

6.3.3. Sensibilitatea

Funcția de sensibilitate S(s) a sistemului automat arată cât de mult se modifică funcția de transfer în regim de urmărire $H_u(s)$ atunci când funcția de transfer a instalației automatizate G(s) îşi schimbă puțin valoarea cu dG(s). Funcția de sensibilitate este definită de relația (6.22).

$$S(s) = \frac{\dfrac{dH_u(s)}{H_u(s)}}{\dfrac{dG(s)}{G(s)}} = \frac{dH_u}{dG}\frac{G}{H_u} \qquad (6.22)$$

Din relațiile (6.21) şi (6.22) rezultă expresia *(6.23)* a *funcției de sensibilitate S(s)*.

$$S(s) = \frac{1}{1+K(s)G(s)} = \frac{1}{1+L(s)} \qquad (6.23)$$

în care *L(s)=K(s)G(s)* este funcția de transfer cu bucla deschisă.

6.3.4. Sensibilitatea complementară

Pe lângă funcția de sensibilitate S(s) se defineşte şi *funcția de sensibilitate complementară T(s)* a sistemului automat din Fig. 6.9 cu ajutorul relației (6.24).

$$T(s) = \frac{K(s)G(s)}{1 + K(s)G(s)} = \frac{L(s)}{1 + L(s)} \tag{6.24}$$

în care L(s)=K(s)G(s) este funcția de transfer cu bucla deschisă.

Din relația (6.24) se observă că pentru traductoarele ideale care au H(s)=1 funcția de sensibilitate complementară este identică cu funcția de transfer (6.21) a sistemului automat.

Se observă că are loc relația (6.25) între cele două sensibilități.

$$S(s) + T(s) = 1 \tag{6.25}$$

6.3.5. Locul rădăcinilor

Dacă se scrie funcția de transfer cu bucla deschisă L(s) în funcție de zerourile z_i şi polii p_i săi,

$$L(s) = k_0 \frac{\prod_i (s + z_i)}{\prod_i (s + p_i)} \tag{6.26}$$

se obține din (6.24) următoarea formă pentru funcția de transfer cu bucla închisă (sensibilitate complementară):

$$T(s) = \frac{k_0 \prod_i (s + z_i)}{\prod_i (s + p_i) + k_0 \prod_i (s + z_i)} \tag{6.27}$$

Se observă că la închiderea buclei, adică la formarea sistemului automat, se întâmplă următoarele lucruri:

1. Zerourile rămân neschimbate.
2. Dacă $k_0 = 0$ polii rămân neschimbați.
3. Dacă $k_0 = \infty$ polii cu bucla închisă sunt egali cu zerourile cu bucla deschisă.

Deci la închiderea buclei şi modificarea lui k_0 de la zero la infinit polii lui T(s) se deplasează din polii lui L(s) în zerourile lui L(s). Reprezentarea grafică a acestei deplasări formează locul rădăcinilor.

Un sistem automat este stabil dacă polii săi sunt situați în semiplanul stâng. Deci locul rădăcinilor ne oferă indicii serioase despre modul în care sistemul automat se îndreaptă spre instabilitate. Satisfacerea anumitor performanțe ale sistemului automat impune anumite valori pentru polii lui T(s), adică plasarea lor în anumite zone ale locului rădăcinilor.

6.3.6. Ieşirea, eroarea si comanda instalației automate

Folosind funcțiile de sensibilitate S(s) şi T(s) se obțin din (6.20) ieşirea, eroarea şi comanda pentru schema bloc din Fig. 6.9

$$Y(s) = T(s)R(s) + S(s)P(s) - T(s)N(s) \tag{6.28}$$

$$E(s) = R(s) - Y(s) - N(s) = S(s)[R(s) - P(s) - N(s)] \tag{6.29}$$

$$U(s) = K(s)E(s) = K(s)S(s)\big[R(s) - P(s) - N(s)\big] = \frac{T(s)}{G(s)}[R(s) - P(s) - N(s)] \tag{6.30}$$

6.3.7. Măsuri ale erorii

Referința R(s), perturbația P(s) şi zgomotul N(s) pot fi generate cu ajutorul *aceluiaşi* model caracterizat de relația (6.31).

$$\nu(s) = W(s)\nu^*(s) \tag{6.31}$$

în care $\upsilon^*(s)$ este transformata Laplace a unui semnal tip,

W(s) – o funcție de transfer proprie fiecărui semnal R(s), P(s) sau N(s).

De exemplu, dacă semnalul tip este impulsul Dirac δ(t), atunci transformata sa Laplace este $\upsilon^*(s)=1$. Pentru W(s)=1/s modelul (6.31) generează un semnal treaptă iar pentru W(s)=1/(as+1) generează un semnal exponențial.

Dacă se consideră separat[143] regimul de urmărire, reglare[144] şi filtrare atunci eroarea (6.29) poate fi exprimată într-un mod foarte general prin relația.

$$E(s) = S(s)W(s)\nu^*(s) \tag{6.32}$$

Pentru cazul $\upsilon^*(s)=1$, analizat mai înainte, eroarea este:

$$E(s) = S(s)W(s) \tag{6.33}$$

în care S(s) şi W(s) sunt funcții de transfer.

O măsură matematică a erorii se poate face cu ajutorul normei H_2 definită de (6.34) sau a normei H_∞ definită de (6.35) .

$$\|e(t)\|_2 = \left(\int_0^\infty |e(t)|^2 dt\right)^{1/2} \tag{6.34}$$

Norma H_2 este egală cu energia semnalului de eroare.

---

[143] Pentru sistemele liniare se poate aplica principiul superpoziției.

[144] Stabilizare.

$$\left\|e(t)\right\|_\infty = \sup_t \left|e(t)\right| \qquad (6.35)$$

Norma H_∞ a erorii (6.35) este egală cu valoarea celui mai mare vârf posibil. Se foloseşte *sup* în loc de *max* pentru că acest vârf poate apare la infinit.

Evaluarea performanţelor se face şi în domeniul frecvenţă şi din această cauză exista norma H_2 şi norma H_∞ a unei funcţii de transfer. Dacă considerăm în (6.28) numai efectul lui P:

$$Y(s) = S(s)P(s) \qquad (6.36)$$

atunci norma H_2 a lui S(s) este

$$\left\|S\right\|_2 = \left(\frac{1}{2\pi}\int_{-\infty}^{\infty}\left|S(j\omega)\right|^2 d\omega\right)^{1/2} \qquad (6.37)$$

şi poate fi interpretată din (6.36) drept valoarea eficace a ieşirii Y a sistemului pentru o intrare P de tip zgomot alb. Norma H_∞ a lui S(s) se defineşte astfel:

$$\left\|S\right\|_\infty = \sup_p \frac{\left\|SP\right\|_2}{\left\|P\right\|_2} \qquad (6.38)$$

Aplicând teorema lui Parceval se obţine o formă echivalentă pentru (6.38):

$$\left\|S\right\|_\infty = \sup_\omega \left|S(j\omega)\right| \qquad (6.39)$$

Ultima relaţie arată că norma H_∞ a lui S(s) este egală cu valoarea de vârf M_s a răspunsului în frecvenţă S(jω) a sistemului.

Proiectarea inginerească a sistemelor automate necesită rezolvarea a două probleme: 1)stabilitatea, 2)asigurarea performanţelor impuse.

6.3.8. Metoda clasică de proiectare

Abordarea clasică a proiectării inginereşti consideră că cele două probleme sunt rezolvate intr-o primă aproximare dacă indicatorii *marginea de amplificare* ***MA*** şi *marginea de fază*[145] ***MF*** au valori care satisfac următoarele condiţii:

$$MA > 2 \quad (6dB), \quad cel\ putin \quad 1{,}6 \quad (4dB) \qquad (6.40)$$

$$60^\circ > MF > 30^\circ \qquad (6.41)$$

. Din păcate există situaţii în care aceşti indicatori nu dau informaţii corecte. Un exemplu va lămurii situaţia.

[145] Rezerva de amplificare, rezerva de fază.

În relația (6.21) să considerăm funcția de transfer (6.42) pentru partea fixă[146] a instalației şi funcția de transfer (6.43) pentru compensatorul PID serie cu filtrare[147].

$$G(s) = \frac{e^{-0,4s}}{(s+1)^4} \qquad (6.42)$$

$$K(s) = K_r\left(1+\frac{1}{T_i s}\right)\frac{T_d s+1}{\alpha T_d s+1} = 1,195(1+\frac{1}{3,23s}+2,3s)\frac{1}{0,2s+1} \qquad (6.43)$$

D

acă notăm cu *L(s)* funcția de transfer a sistemului cu ***bucla deschisă*** definită în *(6.23)*, atunci marginea de amplificare ***MA*** şi marginea de fază ***MF*** pot fi determinate pe diagrama Nyquist a lui *L(jω)* din *Fig. 6.13* .

Marginea de amplificare ***MA*** este inversul distanței de la origine la punctul de intersecție a hodografului[148] lui *L(jω)* cu axa reală şi reprezintă factorul cu care trebuie înmulțit modulul lui *L(jω)* pentru ca sistemul automat să devină instabil[149].

Marginea de fază ***MF*** este unghiul dintre axa reală negativă şi dreapta care uneşte originea cu punctul de intersecție a cercului unitar[150] cu hodograful lui *L(jω).* Marginea de fază reprezintă defazajul suplimentar al lui *L(jω)* pentru ca sistemul automat să devină instabil.

[146] Elementul de execuție, procesul condus şi traductorul.

[147] Pentru a fi realizabil fizic.

[148] Curba funcției de transfer gradată în frecvențe (pulsații). Această reprezentare se mai numeşte şi *loc (hodograf) Nyquist.*

[149] Hodograful lui *L(jω)* să treacă prin punctul de coordonate (-1, j0).

[150] Cercul cu raza R=1.

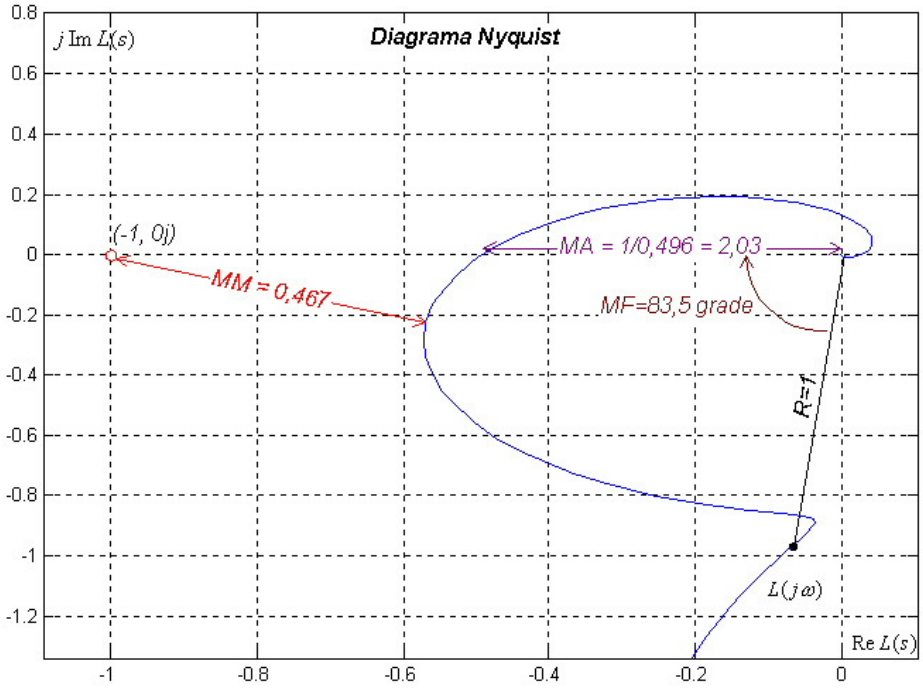


Fig. 6.13 Determinarea marginii de amplificare MA, marginii de fază MF şi a marginii de modul

Pentru exemplul considerat rezultă din *Fig. 6.13* o margine de amplificare *MA=2,03* şi o margine de fază *MF=83,5º*.

Conform regulilor (6.40) şi (6.41) sistemul automat este bine proiectat. Totuşi, se observă în *Fig. 6.13* că hodograful lui *L(jω)* se apropie în mod periculos de *punctul critic (-1, j0)* şi sistemul poate deveni instabil.

6.3.9. Metoda modernă de proiectare

Abordarea modernă în proiectarea inginerească a sistemelor automate consideră un indicator mai bun al robusteții stabilității. Acesta este o distanță minimă, numită *margine de modul* ***MM***, dintre hodograful lui *L(jω)* şi punctul critic.

Marginea de modul ***MM*** este raza ceculeui cu centrul în punctul critic (-1, j0) şi tangent la hodograful lui *L(jω).* Vectorul care unește punctul critic (-1, j0) cu punctul cel mai apropiat de pe hodograful lui *L(jω)* are modulul dat de relația (6.44).

$$MM = \left|1 + L(j\omega)\right|_{\min} \tag{6.44}$$

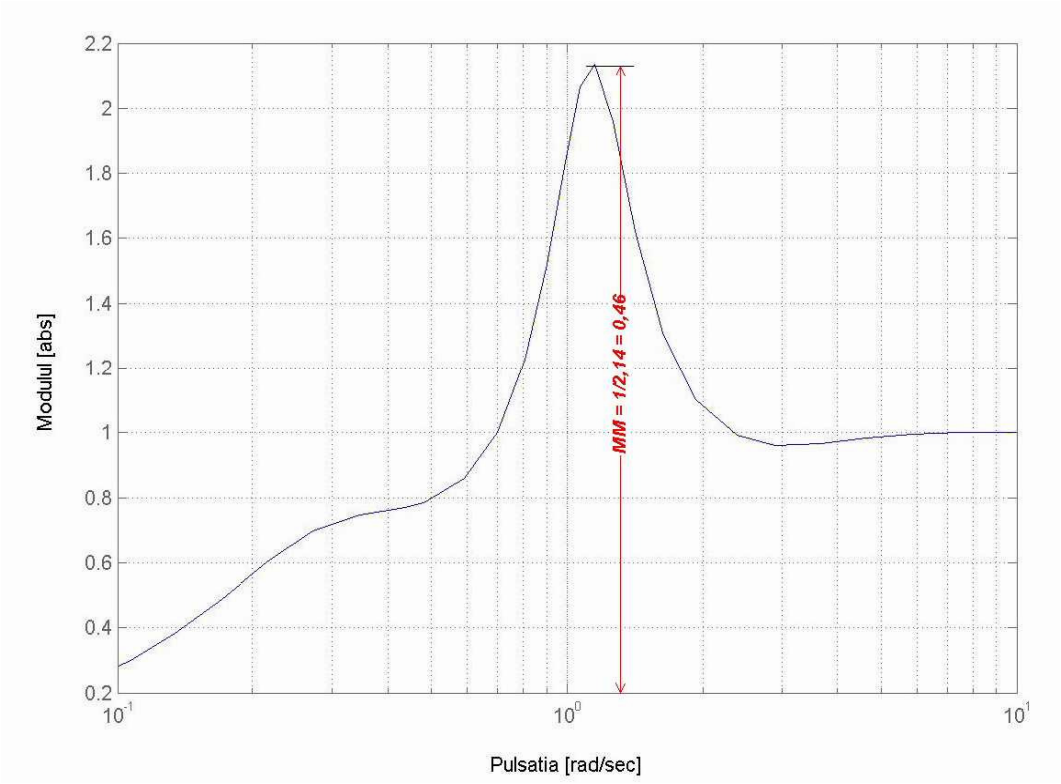


Fig. 6.14 Modulul funcţiei de S(jω) a sistemului automat cu partea fixă (6.42) şi compensatorul PID (6.43)

O nouă definiție (6.45) a marginii de modul ***MM*** rezultă din relațiile *(6.23)* şi (6.44).

$$MM = \frac{1}{\left|S(j\omega)\right|_{\max}} \qquad (6.45)$$

în care $M_s=|S(j\omega)|_{max}$ este valoarea[151] cea mai mare dintre vârfurile modulului lui *S(jω).*

Reducerea sensibilității maxime va conduce la creşterea marginii de modul.

Ca şi în cazul marginii de amplificare sau a marginii de fază există o regulă (6.46) pentru valoarea marginii de modul ***MM*** care asigură robustețea stabilității sistemului automat.

$$MM \geq 0.5\,(-6dB) \quad cel\ putin \quad 0{,}4\,(-8dB) \qquad (6.46)$$

Marginea de modul se calculează uşor cu relația (6.45). Pentru exemplul considerat modulul funcției de sensibilitate este prezentat în *Fig. 6.14*. Valoarea sa maximă este M_s=2,14 şi deci rezultă din (6.45) valoarea ***MM*** = 0,467. Sistemul

[151] Această valoare corespunde normei H_∞ (6.39) a modulului lui *S(jω)* şi se notează cu M_s.

automat nu îndeplineşte condiţia (6.46) şi nu este stabil robust. Această concluzie coincide şi cu situaţia vizibilă în *Fig. 6.13*.

Marginea de modul este foarte importantă şi pentru că face legătura cu criteriul cercului de stabilitate al lui Popov pentru sistemele neliniare.

Condiţia de robusteţe (6.46) implică şi realizarea condiţiilor de robusteţe (6.40) şi (6.41) După cum am văzut din exemplul prezentat reciproca nu este valabilă.

Creşterea unor performanţe ale sistemului automat implică micşorarea robusteţii stabilităţii sale. Marginea de modul este cel mai bun indicator al faptului că o anumită limită a robusteţii, de exemplu cea indicată de relaţia (6.46), nu este depăşită.

6.3.10. Criteriu de stabilitate robustă

Pericolul datorat unui grad de incertitudine în cunoaşterea instalaţiei G(s) poate fi aditiv sau multiplicativ. Riscul aditiv a(jω) este definit astfel:

$$G(j\omega) = G_m(j\omega) + a(j\omega) \tag{6.47}$$

în care $G_m(j\omega)$ este modelul instalaţiei.

Pentru înlesnirea proiectării se introduce o constantă dependentă de frecvenţă a(ω) astfel încât:

$$\left|a(j\omega)\right| \le a(\omega) \tag{6.48}$$

Asemănător se defineşte riscul multiplicativ l(jω):

$$G(j\omega) = G_m(j\omega)\left[1 + l(j\omega)\right] \tag{6.49}$$

în care:

$$\left|l(j\omega)\right| \le l(\omega) \tag{6.50}$$

şi apare din nou o constantă l(ω) dependentă de frecvenţă introdusă pentru specificarea sarcinilor urmărite la proiectare.

Deoarece ambele descrieri trebuie să fie echivalente este necesar ca:

$$l(\omega) = \frac{a(\omega)}{\left|G_m(j\omega)\right|} \tag{6.51}$$

Dacă se cunoaşte o margine superioară l(ω) (6.50) a incertitudinii modelării instalaţiei automatizate se poate stabilii un criteriu de stabilitate robustă mai bun decât criteriile (6.40), (6.41) sau (6.46). In acest scop se consideră un exemplu în care o instalaţie este modelată cu ajutorul funcţiei de transfer (6.52).

$$G(s) \approx G_m(s) = \frac{e^{-0.5s}}{1+s} \tag{6.52}$$

în care $G_m(s)$ este funcţia de transfer a modelului adoptat pentru instalaţie.

Compensatorul erorii este de tip proporţional şi are valoarea K(s)=2.

Traductorul sistemului automat prezintă un risc tehnic de 10% până la 1 rad/s. Pentru frecvenţe mai mari riscul creşte liniar până la 20 rad/s, iar după aceea rămâne constant şi egal cu 100%. Această descriere a riscului este multiplicativă. Pentru folosirea diagramei Nyquist se determină riscul aditiv cu ajutorul relaţiei (6). In *Fig. 6.15* se prezintă grafic funcţia de transfer a sistemului automat cu bucla deschisă. Riscul aditiv apare sub forma unor cercuri suprapuse.

Criteriul de stabilitate Nyquist arată că dacă funcţiile de transfer posibile ale instalaţiilor modelate au acelaşi număr de poli în semiplanul drept şi dacă banda de cercuri nu include punctul (-1,0) atunci sistemul automat este stabil robust. Cu alte cuvinte stabilitatea este garantată dacă distanţa de la punctul (-1,0) la un punct al graficului K(jω)G(jω) este mai mică decât riscul tehnic reprezentat de raza cercului:

$$\left|1+K(j\omega)G_m(j\omega)\right| > \left|K(j\omega)G_m(j\omega)\right| l(\omega) \qquad (6.53)$$

Din *Fig. 6.15* se observă că sistemul automat nu are risc de stabilitate, adică este stabil robust, deoarece relaţia (6.53) este satisfăcută.

Dacă se consideră sensibilitatea complementară $T_m(s)$ pentru instalaţia nominală,

$$T_m(s) = \frac{K(s)G_m(s)}{1+K(s)G_m(s)} \qquad (6.54)$$

atunci criteriul de stabilitate robustă rezultă din (6.53) şi este:

$$\left|T_m(j\omega)\right| l(\omega) < 1 \qquad (6.55)$$

pentru toate valorile lui ω.

Un alt exemplu de apreciere al stabilităţii robuste îl constituie cazul modelării timpului mort T_m al modelului instalaţiei automatizate cu ajutorul aproximaţiei Pade de ordinul doi (6.80).

$$e^{-T_m s} \cong V(s) = \frac{12-6T_m s+T_m^2 s^2}{12+6T_m s+T_m^2 s^2} \qquad (6.56)$$

Funcţia de transfer a procesului este (6.79).

$$G(s) = K_f \frac{e^{-T_m s}}{(T_f s+1)^n} \qquad (6.57)$$

Funcţia de transfer a modelului procesului este (6.58).

$$G_m(s) = K_f \frac{V(s)}{(T_f s+1)^n} \qquad (6.58)$$

Înlocuind pe (6.80), (6.79) şi (6.58)în (6.49) se poate calcula riscul multiplicativ (6.59).

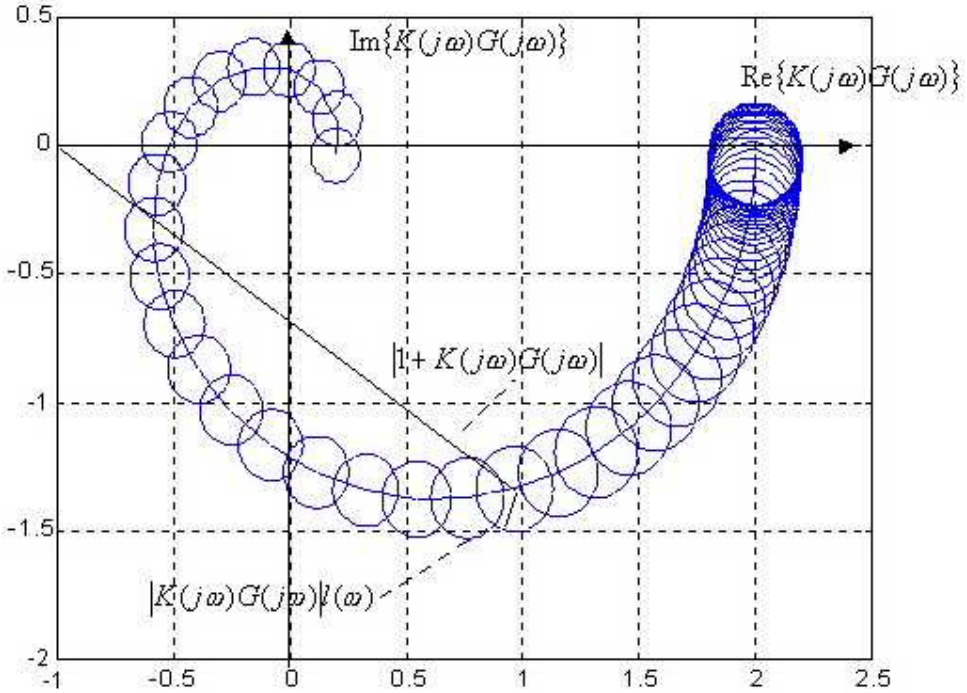


Fig. 6.15 Diagrama Nyquist cu parametrii riscului de stabilitate.

$$l(j\omega) = e^{-j\omega T_m} \frac{1}{V(j\omega)} - 1 \qquad (6.59)$$

Dacă avem un model pentru proces, de exemplu $G_m(s)$ dat de (6.58), se poate proiecta compensatorul K(s) şi apoi cu ajutorul relaţiilor (6.54) şi (6.59) se poate verifica condiţia de stabilitate robustă (6.55). Un exemplu va clarifica lucrurile. Pentru modelul procesului descris de relaţiile (6.80) şi (6.58) avem următoarele valori pentru parametrii:

$$K_f = 1\,; \quad T_m = 5[s]\,; \quad T_f = 10[s]\,; \quad n = 1 \qquad (6.60)$$

Acordarea Ziegler – Nichols a regulatorului conduce la următorii parametri ai compensatorului erorii:

$$K_r = 3 \quad ; \quad T_i = 12{,}5[s] \quad ; \quad T_d = 2{,}5[s] \qquad (6.61)$$

Funcţia de transfer a regulatorului PID va fi:

$$K(s) = K_r\left(1 + \frac{1}{T_i s} + T_d s\right) = 3\left(1 + \frac{1}{12{,}5s} + 1{,}5s\right) \qquad (6.62)$$

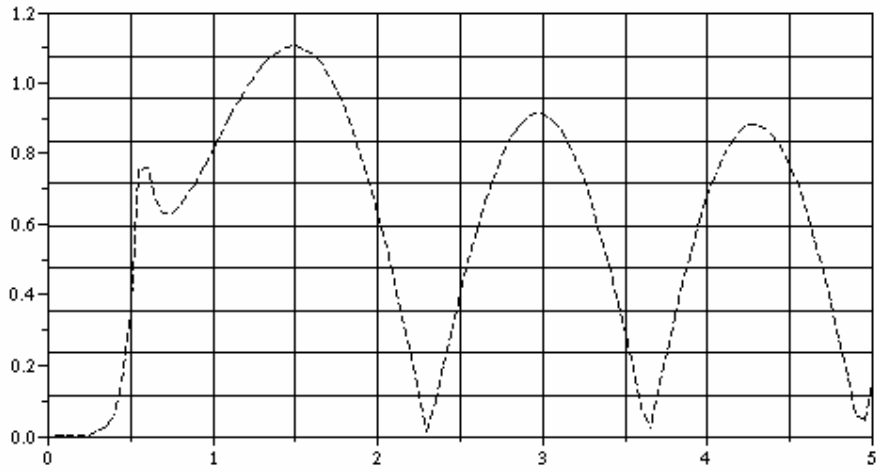


Fig. 6.16 Variaţia condiţiei (6.55) în funcţie de ω pentru valorile (6.60) şi (6.61)

Cu valorile numerice (6.60) şi (6.61) condiţia (6.55) are valoarea din *Fig. 6.16*. Se observa ca ea nu este îndeplinită, depăşind valoarea 1 cu toate că regulatorul a fost acordat Ziegler-Nichols.

6.3.11. Criteriu de performanţe robuste

Riscul tehnic la sistemele automate implică în afară de riscul instabilităţii şi riscul deteriorării performanţelor. Una dintre cele mai importante performanţe apreciază modul în care sistemul automat înlătură sau atenuează efectul perturbaţiilor. Pentru perturbaţii diferite eroarea poate fi exprimată, într-un mod foarte general, cu ajutorul relaţiilor (6.32) şi (6.33). Deoarece se consideră că perturbaţia este unitară rezultă că răspunsul sistemului automat la perturbaţie, adică eroarea, trebuie să fie mai mic decât unu. Folosind norma H_∞ aplicată relaţiei (6.33) rezultă condiţia de performanţă nominală:

$$\|SW\|_\infty < 1 \qquad (6.63)$$

Pentru o performanţă robustă relaţia (6.63) trebuie să fie satisfăcută pentru cazul cel mai nefavorabil al funcţiei de transfer al procesului din instalaţie. Deci:

$$\|SW\|_\infty = \sup_\omega |S(j\omega)W(j\omega)| < 1 \qquad (6.64)$$

în care *sup* – supremum înseamnă că se consideră valoarea cea mai mare a modulului funcţiei de sensibilitate S(jω) ponderate cu W(jω), pentru toate valorile lui ω.

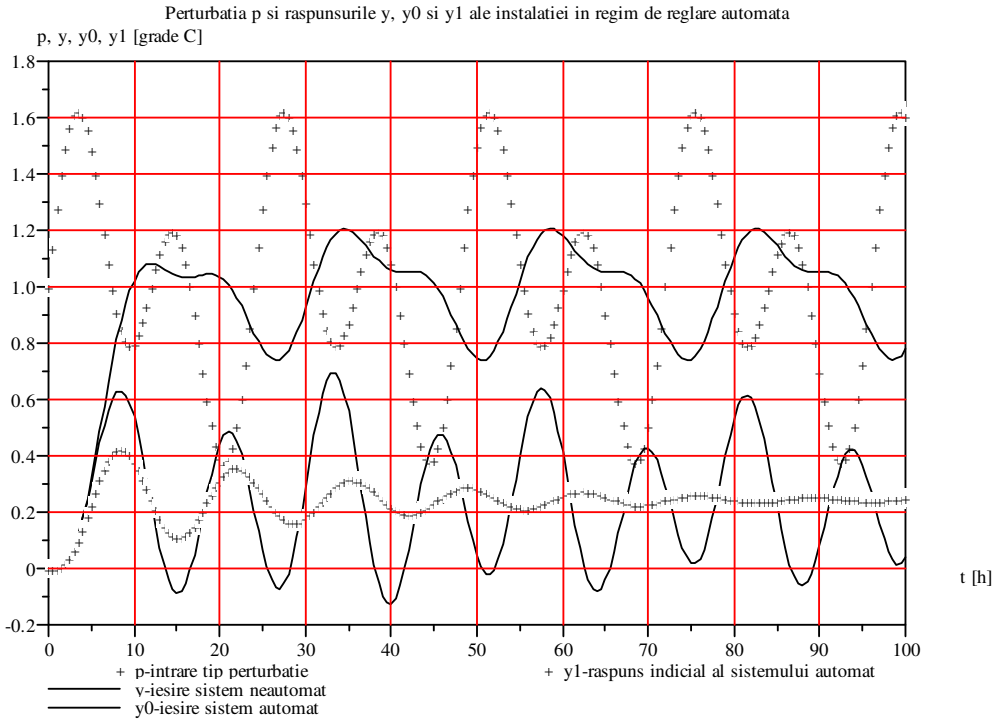


Fig. 6.17 Analiza în domeniul timp a sistemului neautomat şi sistemului automat cu regulator tip P (proporţional) în regim de reglare.

Se observă în *Fig. 6.15* că pentru orice frecvenţă funcţia de transfer în buclă deschisă *GK* se va găsi în interiorul discurilor care reprezintă regiunea de incertitudine. Din această cauză există următoarea relaţie pentru toate valorile posibile ale lui G:

$$\left|1+G(s)K(s)\right| \geq \left|1+G_m(s)K(s)\right| - \left|G_m(s)K(s)\right|l \qquad (6.65)$$

Din definiţia *(6.23)* a funcţiei de sensibilitate S rezultă:

$$\left|S\right| = \left|\frac{1}{1+GK}\right| \leq \frac{\left|S_m\right|}{1-\left|T_m\right|l} \qquad (6.66)$$

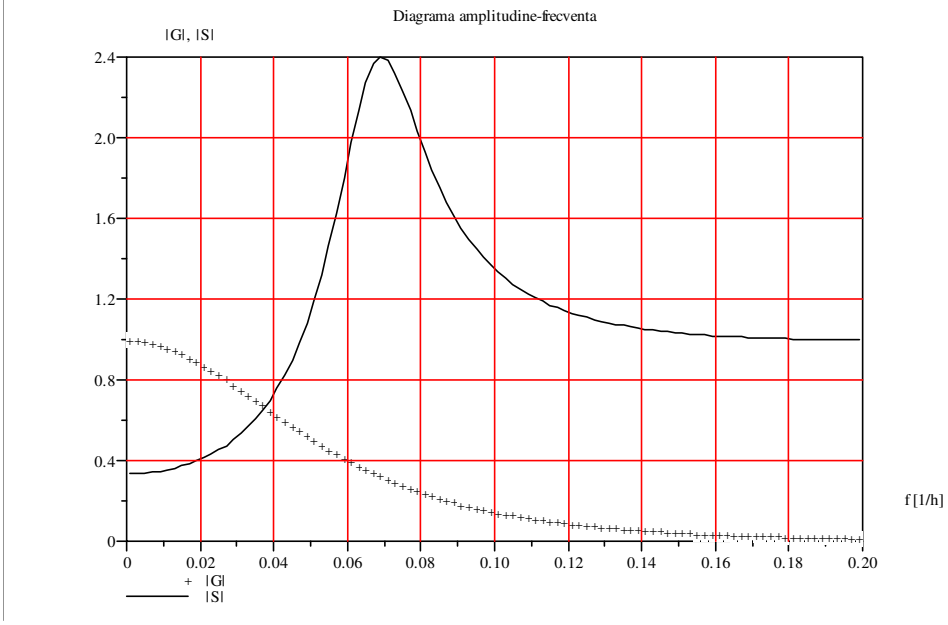


Fig. 6.18 Analiza în domeniul frecvență a sistemului neautomat şi a sistemului automat cu regulator tip P (proporțional).

Condiția de *performanță robustă* (6.64) pentru toate pulsațiile ω devine:

$$\frac{|S_m W|}{1-|T_m|l} < 1 \qquad (6.67)$$

sau

$$|T_m l| + |S_m W| < 1 \qquad (6.68)$$

Se observă că condiția de performanță robustă (6.68) implică condiția de stabilitate robustă (6.55) şi condiția de performanță nominală (6.63). Se observă că îmbunătățirea stabilității robuste provoacă o deteriorare a performanței nominale şi invers. Pe de altă parte funcția de sensibilitate S(s) trebuie să îndeplinească şi condiția integrală (6.69) a lui Bode.

$$\int_0^\infty \ln|S(j\omega)|d\omega = 0 \qquad (6.69)$$

Considerând relațiile (6.25), (6.68) şi (6.69) şi impunând diferite condiții lui S(s) şi T(s) se pot lua în considerare diferite performanțe referitoare la stabilitatea robustă, eroarea staționară, suprareglarea, durata procesului tranzitoriu, banda de trecere în buclă închisă şi la eliminarea perturbațiilor, a zgomotului de măsurare, a deficiențelor de modelare şi a saturării elementului de execuție. De exemplu, pentru sistemul automat standard din Fig. 6.9 T(s) definit de relația

(6.54) reprezintă funcţia de transfer intrare – ieşire. Dacă această funcţie de transfer este de ordinul doi, aşa cum se consideră la evaluarea performanţelor standard, atunci vârful M_t al diagramei Bode a modulului lui T(s) are valoarea:

$$M_t = \sup_{\omega} |T_m(j\omega)| = \frac{1}{2\zeta\sqrt{(1-\zeta^2)}} \qquad (6.70)$$

în care ζ este fracţiunea de amortizare critică.

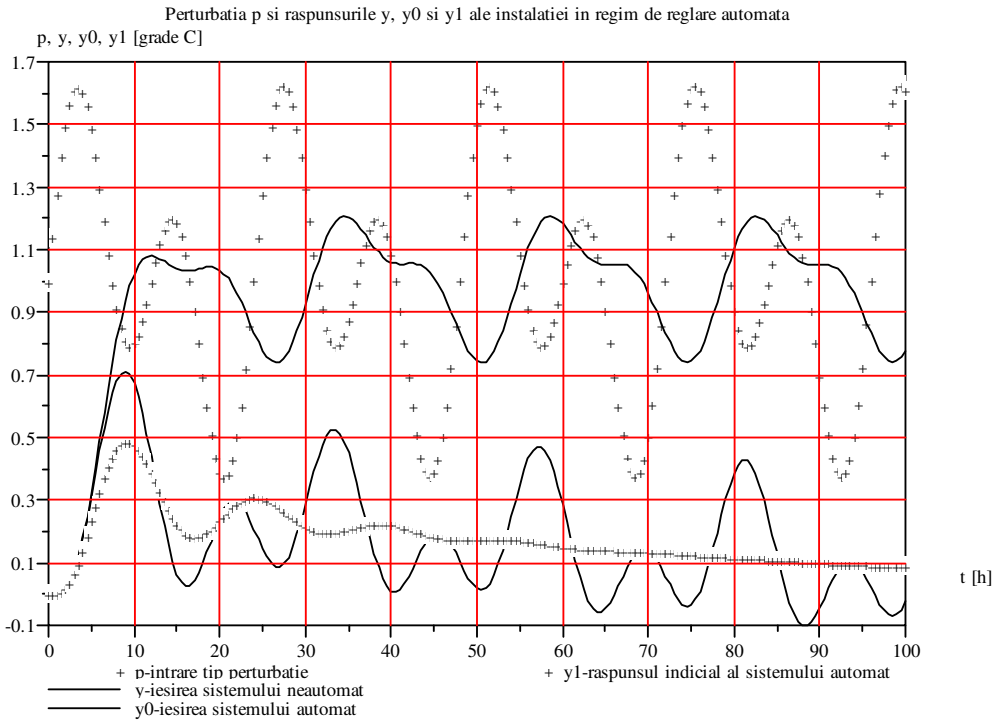


Fig. 6.19 Analiza în domeniul timp a sistemului neautomat şi a sistemului automat cu regulator tip PI (proporţional integral) în regim de reglare.

Suprareglarea răspunsului indicial, o performanţă importantă pentru sistemele automate, depinde şi ea de ζ:

$$\sigma = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \qquad (6.71)$$

Din (6.55), (6.70) şi (6.71) se poate stabilii o legătură între robusteţea stabilităţii şi indicatorii de performanţă standard. Câteva valori folosite frecvent sunt următoarele:

Tab. 6.3

M_t =	1,1	…	1,5
ζ =	0,54	…	0,36
σ =	0,13	…	0,29

Recunoaşterea faptului că la proiectarea şi exploatarea sistemelor automate este necesară considerarea riscului tehnic a condus la apariţia mai multor metode de proiectare, în special în domeniul frecvenţă, pe baza conceptelor prezentate în această lucrare.

6.3.12. Compensarea perturbaţiilor

Relaţiile definite până acum sunt suficiente pentru determinarea performanţelor unui sistem automat în regim de reglare. Se pot face câteva observaţii.

În regim de reglare R(s)=0 şi N(s)=0. Cu aceste condiţii relaţiile (6.28), (6.29) şi (6.30) cu ajutorul cărora se pot calcula ieşirea, eroarea şi comanda sistemului automat devin:

$$Y(s) = S(s)P(s) \tag{6.72}$$

$$E(s) = R(s) - Y(s) - N(s) = -S(s)P(s) \tag{6.73}$$

$$U(s) = K(s)E(s) = -K(s)S(s)P(s) = -\frac{T(s)}{G(s)}P(s) \tag{6.74}$$

Rolul principal în determinarea performanţelor sistemului automat în regim de reglare îl joacă funcţia de sensibilitate S(s) definită în domeniul frecvenţă

Regimul ideal de reglare presupune înlăturarea totală a efectului perturbaţiilor. Din relaţia (6.72) rezultă S(s)=0, condiţie ce poate fi îndeplinită numai aproximativ dacă în (6.23) funcţia de transfer a compensatorului are o valoare foarte mare (K(s)=∞). În această situaţie L(s)=K(s).G(s) devine foarte mare şi numai este îndeplinită condiţia Nyquist de stabilitate a sistemului automat deoarece hodograful L(jω) înconjoară punctul de coordonate (-1, j0).

Se poate îndeplini aproximativ condiţia S(s)=0 numai pentru frecvenţele joase. Datorită relaţiei (6.69) va creşte în acest caz valoarea lui S(s) la frecvenţele înalte. Pentru îndeplinirea condiţiei de stabilitate robustă este necesar ca Mt să nu fie mai mare decât 2.

În loc de o bandă de frecvenţe pentru care sensibilitatea trebuie să fie nulă se pot alege câteva frecvenţe discrete la care S(s) să aibă valori foarte mici.

Toate performanţele definite până acum sunt valabile numai pentru domeniul în care sistemul automat este liniar. Dacă S(s) este foarte mic atunci din (6.25) rezultă că T(s) este foarte mare şi este posibil să rezulte din (6.74) comenzi foarte mari care nu pot fi realizate fizic de către elementul de execuţie. Cu alte cuvinte sistemul automat intră într-un domeniu neliniar de funcţionare în care toate relaţiile stabilite nu mai sunt valabile.

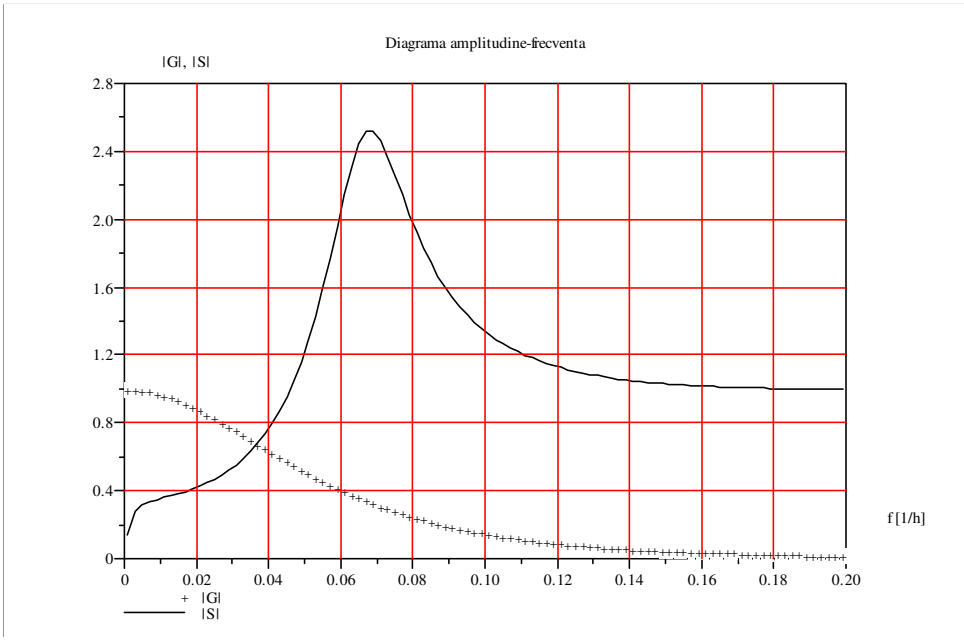


Fig. 6.20 Analiza în domeniul frecvenţă a sistemului neautomat şi a sistemului automat cu regulator tip PI (proporţional integral).

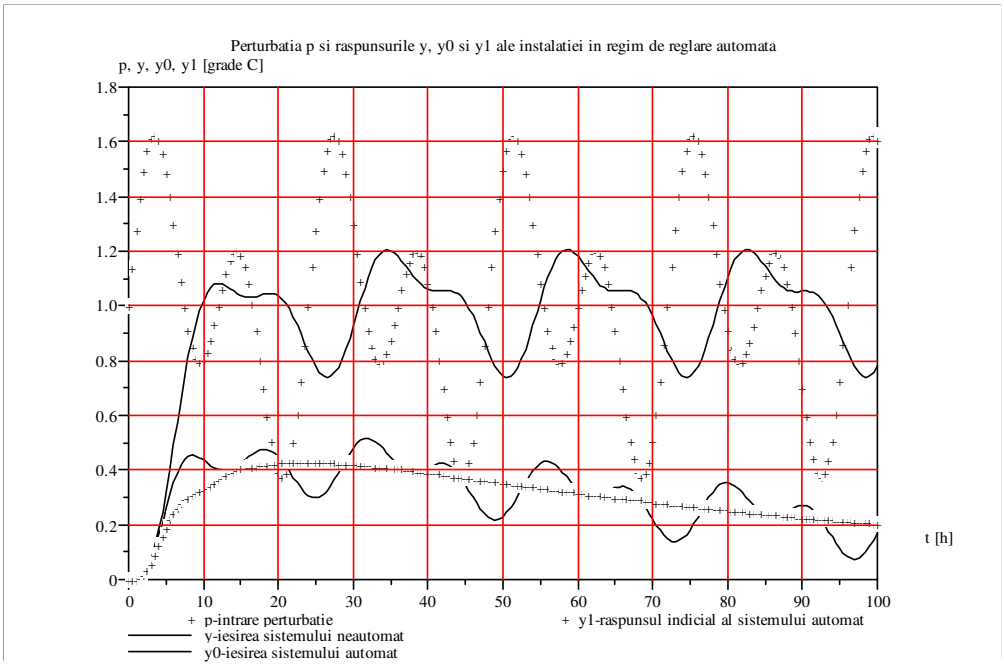


Fig. 6.21 Analiza in domeniul timp a sistemului neautomat si a sistemului automat cu regulator PID (proporţional integral derivativ) in regim de reglare

Considerând relaţiile (6.25), şi (6.69) şi impunând diferite condiţii lui S(s) şi T(s) se pot lua în considerare diferite performanţe din domeniul timp referitoare la stabilitatea robustă, eroarea staţionară, suprareglarea, durata procesului tranzitoriu, banda de trecere în buclă închisă şi la eliminarea perturbaţiilor, a zgomotului de măsurare, a deficienţelor de modelare şi a saturării elementului de execuţie. Exemplele prezentate în continuare folosesc programe Scilab care pot fi descărcate de la adresa www.geocities.com/larionescu/.

Exemplul 1. Pentru sistemul automat standard din Fig. 6.9 funcţia de sensibilitate complementară *T(s),* definită de relaţia (6.24), reprezintă funcţia de transfer intrare – ieşire. Dacă această funcţie de transfer este de ordinul doi, aşa cum se consideră la evaluarea performanţelor standard, atunci vârful M_t al diagramei Bode a modulului lui *T(s)* are valoarea (6.70)

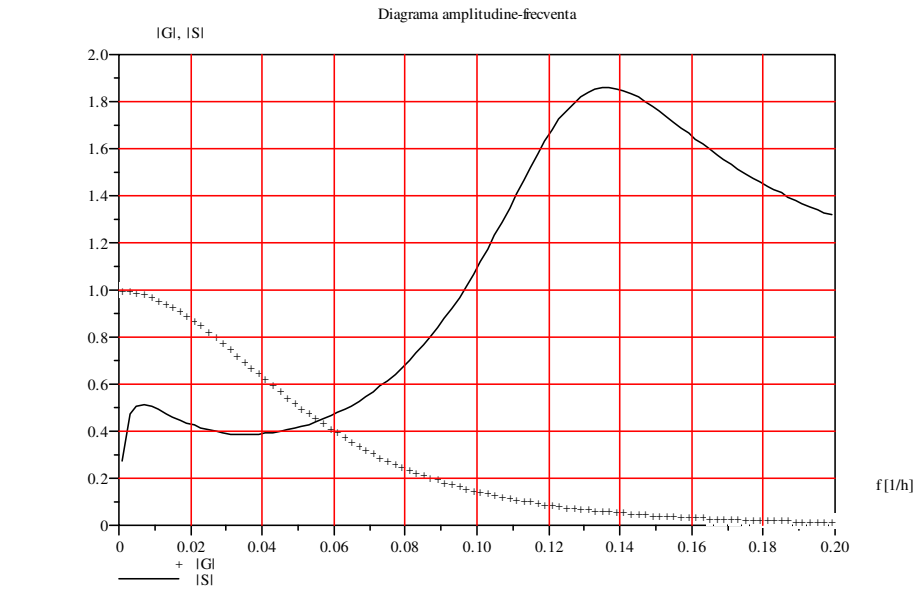


Fig. 6.22. Analiza în domeniul frecvenţă a sistemului neautomat şi a sistemului automat cu regulator tip PID (proporţional integral derivativ)

Suprareglarea răspunsului indicial, o performanţă importantă pentru sistemele automate, depinde şi ea de ζ şi este dată de (6.71).

Din (6.70) şi (6.71) se poate stabilii o legătură între robusteţea stabilităţii şi indicatorii de performanţă standard. Câteva valori folosite frecvent sunt date în Tab. 6.3.

Exemplul 2. Să considerăm o instalaţie de încălzire automatizată pentru o clădire cu inerţie termică mică, Fig. 6.8, având următoarea funcţie de transfer a procesului:

$$G(s) = \frac{1}{(1+2s)^4} \qquad (6.75)$$

în care constanta de timp este exprimată în ore.

Celelalte blocuri din Fig. 6.8 au funcţiile de transfer următoare: *A=G, B=C=D1=D2=0* şi *F=H=1*.

Perturbaţia datorată temperaturii exterioare are următoarea expresie:

$$p(t) = 1 + \sin(0{,}262t) + 0{,}4\sin(0{,}5236t) \qquad (6.76)$$

Perturbaţia instalaţiei automatizate o constituie variaţia temperaturii exterioare care are o componentă continuă, o componentă sinusoidală cu perioada de 24 ore (pulsaţia 0,262) şi o altă componentă sinusoidală cu perioada de 12 ore (pulsaţia 0,523).

Referinţa *r(t)* şi zgomotul *n(t)* sunt nule.

Pentru început am considerat că regulatorul de temperatură este de tip proporţional cu constanta de proporţionalitate $K_r=3$. În Fig. 6.17 se prezintă analiza în domeniul timp a sistemului automat în regim de reglare. Din răspunsul indicial se observă că regulatorul proporţional este acordat aproximativ la sfert (amortizarea egală cu 1/4). Eroarea indicială staţionară este 0,25. Conform recomandărilor curente sistemul automat ar trebuii să funcţioneze satisfăcător, adică să atenueze efectul perturbaţii asupra ieşirii sistemului automat. Din Fig. 6.17 reiese însă că efectul perturbaţii sete mai mare la sistemul automat decât la sistemul neautomat. Explicaţia rezultă din analiză din domeniul frecvenţă prezentată în *Fig. 6.18*. Se observă că componenta sinusoidală a perturbaţiei (6.76) cu pulsaţia 0,523 rad/oră sau frecvenţa 0,0832 cicli/oră este amplificată puternic de sensibilitatea *S* a sistemului automat conform relaţiei (6.72). Vârful sensibilităţii este 2,4 şi corespunde unei margini de modul ***MM***=0,417 adică în limitele acceptabile de robusteţe ale stabilităţii specificate de criteriul (6.46).

S-a încercat îmbunătăţirea funcţionării instalaţiei automatizate cu ajutorul unui regulator proporţional integral acordat aproximativ la sfert şi având următoarea funcţie de transfer:

$$K(s) = 2(1 + \frac{1}{50s}) \qquad (6.77)$$

în care constanta de timp integral este măsurată în ore.

Analiza în domeniul timp şi în domeniul frecvenţă este prezentată în Fig. 6.19 şi *Fig. 6.20*.Se observă că eroarea indicială staţionară devin în acest caz zero, o îmbunătăţire semnificativă, dar sensibilitatea sistemului automat are în continuare un vârf în apropierea frecvenţei de 0,0832 cicli/oră corespunzătoare perioadei de 12 ore a perturbaţiei. De date aceasta sistemul automat nu mai este stabil robust conform criteriului marginii de modul (6.46).

În final am folosit pentru reglarea automată a instalaţiei un regulator PID cu funcţia de transfer:

$$K(s) = 2(1 + \frac{1}{50s} + 10{,}25s)$$

(6.78)

Analiza în domeniul frecvență din Fig. 6.22 arată că vârful sensibilității sistemului automat s-a îndepărtat mult de frecvențele perturbației care este de data aceasta compensată puternic, după cum se vede şi în Fig. 6.21. Cu un vârf al sensibilității mai mic decât 2 sistemul este data aceasta stabil robust.

Al doilea exemplu ilustrează foarte bine importanța analizei în domeniul frecvență a sistemelor automate destinate, în special, să compenseze perturbațiile periodice.

6.3.13. Exemplu de proiectare

Se prezintă un exemplu de utilizare a programului ***ControlKit*** de proiectare a sistemelor automate liniare. Instalarea şi utilizarea programului este prezentată în paragraful 10.3. Modul său de utilizare se obține alegând opțiunea ***?/Scilab Help/Graphicss based tool for SISO system design*** din meniul principal.

O modalitate rapidă de acomodare cu programul constă în proiectarea unui sistem automat urmărind etapele din meniul principal. Rezultă un sistem automat cu partea fixă tip Strejc de ordin 3 cu timp mort şi regulator tip PID ideal cu filtrarea componentei derivative. Proiectarea unui sistem de reglare automată constă, pe scurt, în determinarea compensatorului K(s) atunci când se cunoaşte instalația G(s).Sistemul automat proiectat trebuie să fie verificat din punct de vedere al robusteți stabilității şi performanțelor.

1. Construirea funcțiilor de transfer G(s) şi K(s).

În urma identificării procesului rezultă următoarea funcție de transfer:

$$G(s) = K_f \frac{e^{-T_m s}}{(T_{pf} s + 1)^n} = 2\frac{e^{-s}}{(s+1)^3} \qquad (6.79)$$

După cum se vede $K_f = 2$, $T_{pf} = 1$, numărul polilor n=3 şi $T_m = 1$

Timpul mort T_m al modelului instalației automatizate este aproximat automat de program cu ajutorul aproximației Pade de ordinul doi:

$$e^{-T_m s} \cong \frac{12 - 6T_m s + T_m^2 s^2}{12 + 6T_m s + T_m^2 s^2} = \frac{12 - 6s + s^2}{12 + 6s + s^2} \qquad (6.80)$$

Compensatorul PID ideal cu filtrarea componentei derivative are funcția de transfer:

$$K(s) = K_r\left(1+\frac{1}{T_i s}+\frac{T_d s}{\alpha T_d s+1}\right) = K_r(1+\frac{1}{s}+\frac{0.25s}{0.125s+1}) = K_r\frac{2.66+3s+s^2}{8s+s^2} \tag{6.81}$$

în care T_i =1, T_d =0,25 iar α are valoarea 0,5.
Constantele de timp au fost alese după regula:

$$T_i = T \tag{6.82}$$

$$T_d = 0{,}25T_i \tag{6.83}$$

Funcția de transfer L(s)=K(s)G(s) cu bucla deschisă este determinată sub forma poli-zerouri:

$$L(s) = K(s)G(s) = k_0\frac{\prod_i (s+z_i)}{\prod_i (s+p_i)} \tag{6.84}$$

Polii şi zerourile sunt calculate de către program cu ajutorul relațiilor (6.79), (6.80) şi (6.81). Constanta de proporționalitate k_0 va fi determinată de către program în etapele următoare.

La sfârşitul acestei etape de proiectare programul reprezintă grafic în Fig. 6.23 şi Fig. 6.24, răspunsul indicial şi diagrama Bode a instalației G(s). Acestea sunt necesare pentru a putea verifica identificarea şi pentru a compara performanțele instalației neautomatizate cu cele ale instalației automatizate.

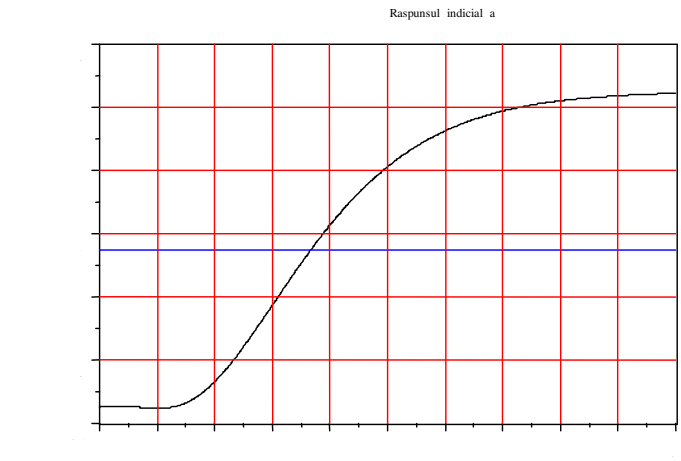


Fig. 6.23 Verificarea identificării în domeniul timp

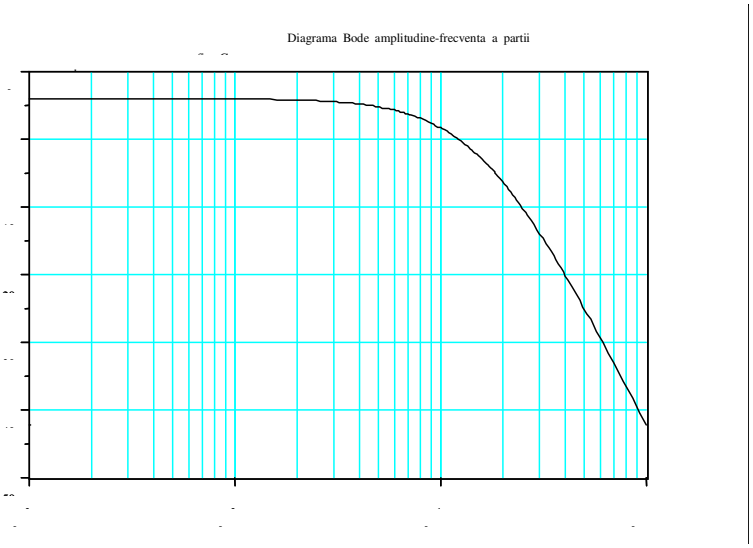


Fig. 6.24 Verificarea identificării în domeniul frecvență

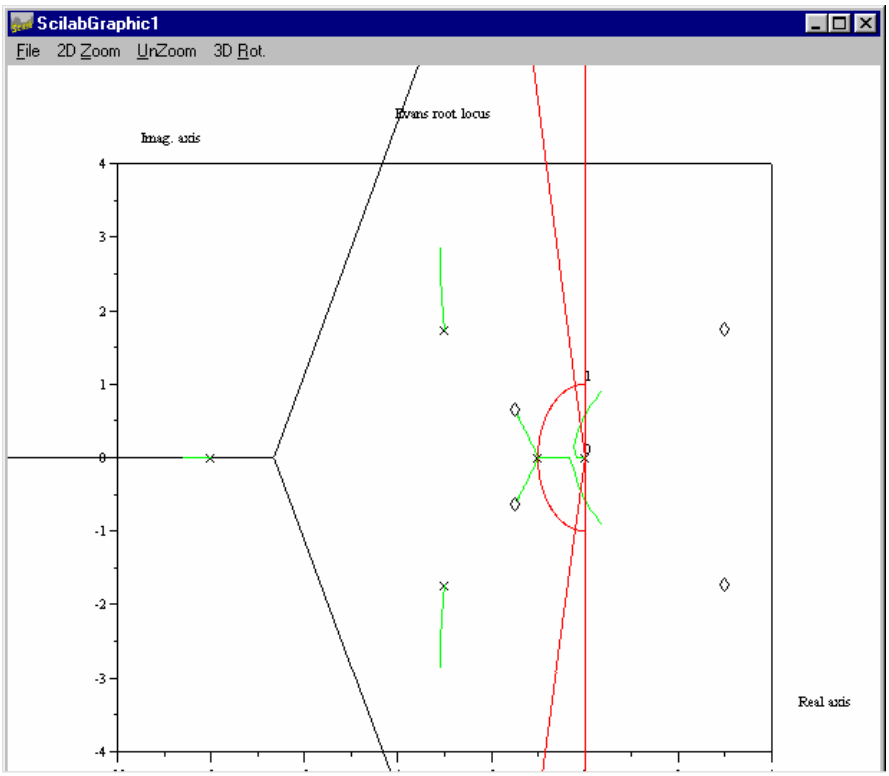


Fig. 6.25 Locul rădăcinilor

1. Locul rădăcinilor

Locul rădăcinilor funcției de transfer L(s) cu bucla deschisă în funcție de k_0 este prezentat în Fig. 6.25. Se delimitează domeniul performanțelor și sunt trasate asimptotele către care tind laturile locului atunci când k_0 tinde către infinit. Polii lui L(s) sunt marcați cu x iar zerourile cu un romb.

2. Constantele k0 şi Kr

Se selectează cu mauseul stâng punctul de intersecție al ramurii locului cu limita domeniului performanțelor. În fereastra principală Scilab apare punctul selectat, valoarea lui k_0 şi a constantei de proporționalitate a compensatorului K_r

$$p_0 = -0{,}083 + 0.492\,j \qquad (6.85)$$

$$k_0 = 1{,}583 \qquad (6.86)$$

$$K_r = 0{,}263 \qquad (6.87)$$

3. Marginea de fază şi marginea de amplificare

Programul desenează diagrama Nyquist a funcției de transfer cu bucla deschisă. Se observă în Fig. 6.26 cercul cu raza unitate necesar pentru a determina rezerva de fază. În fereastra principală Scilab programul afişează marginea de fază, modulul lui L(jω) pentru faza de 180 grade şi marginea de modul.

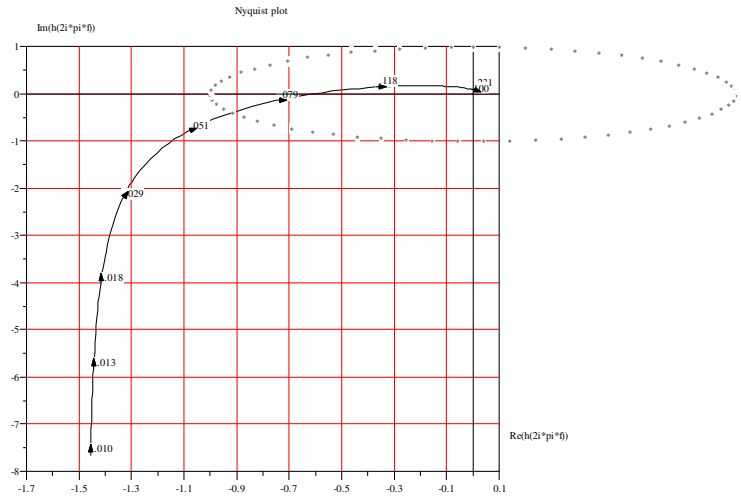


Fig. 6.26 Diagrama Nyquist

$$MFaza = -20{,}08 \quad grade \tag{6.88}$$

$$L180 = 0{,}653 \tag{6.89}$$

$$MAmp = \frac{1}{L180} = 1{,}529 \tag{6.90}$$

Se observă că sistemul automat nu este stabil robust.

4. Sensibilitatea complementară T

Programul determină sensibilitatea complementară T(s) sub formă simbolică şi o afişează în fereastra principală Scilab. În acest caz T(s) este şi funcţia de transfer a sistemului automat în regim de urmărire.

5. Sensibilitatea S şi răspunsul indicial

În Fig. 6.27 se prezintă graficele determinate de program pentru răspunsurile indiciale (la o intrare treaptă) ale ieşirii y(t) şi comenzii u(t). Se observă că se respectă foarte bine performanţa impusă de acordare Ziegler – Nichols (la sfert) şi tradusă în domeniul performanţelor prin condiţia ζ=0,22. Cu toate acestea sistemul automat nu este stabil robust.

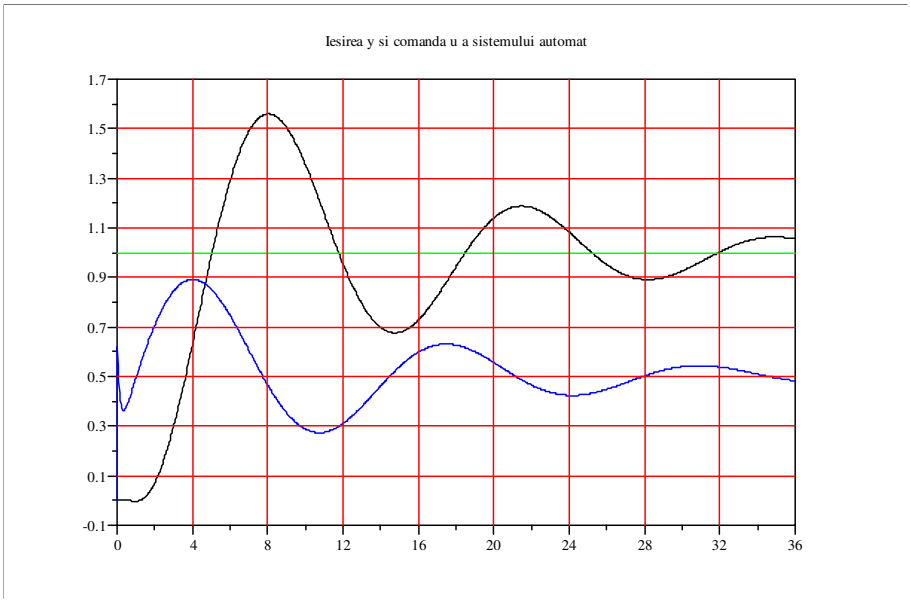


Fig. 6.27 Ieşirea şi comanda sistemului automat

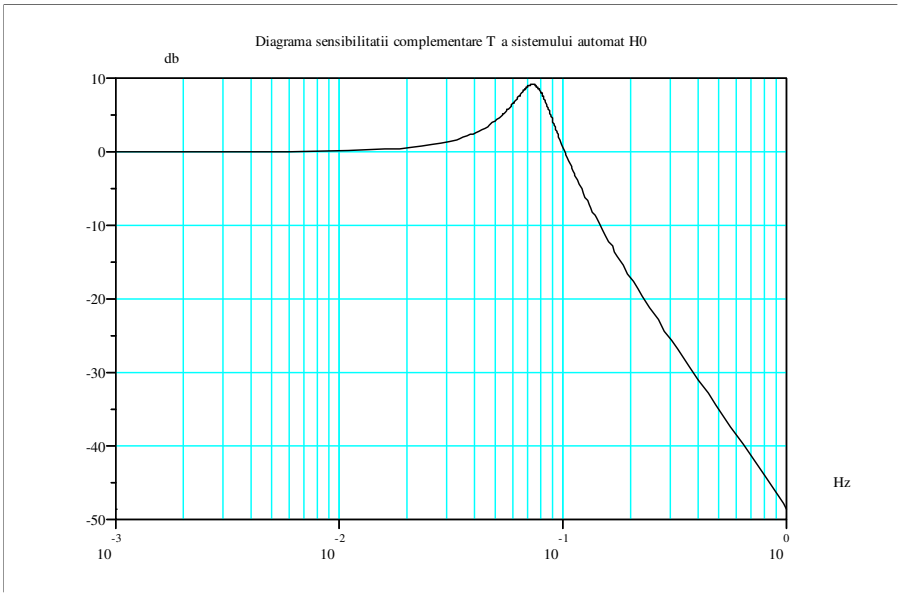


Fig. 6.28 Sensibilitatea complementară

Diagrama Bode a sensibilității complementare T(s) din *Fig. 6.28* permite o comparație cu instalația neautomatizată din Fig. 6.24. Se observă că în ambele cazuri banda de trecere este de aproximativ 0,1 Hz. În consecință ambele sisteme, neautomat şi automat, au aceiaşi viteză. Tot din această observație rezultă şi faptul că mărimea de comandă nu are vârfuri accentuate.

Din diagrama Bode a sensibilității S(s), *Fig. 6.29*, rezultă că marginea de modul este:

$$MModul = -9 \quad dB \tag{6.91}$$

adică nu este îndeplinită condiția de robustețe.

Comparând *Fig. 6.28* cu *Fig. 6.29* se observă că este îndeplinită relația:

$$S(s) + T(s) = 1 \tag{6.92}$$

6.3.14. Exemplu de simulare

O buclă cu regulator ideal PID este simulată cu ajutorul unui program Scilab/Scicos prezentat în Fig. 6.30. Programul poate fi descărcat de la adresa internet www.geocities.com/larionescu/. Blocurile din schemă sunt *obiecte* de program. Daca în fereastra *Scicos* se alege meniul *Object* se pot executa diferite comenzi asupra blocului care va fi selectat cu mauseul stâng.

Comanda:

Object / Documentation

oferă o documentare asupra obiectului selectat.

Comanda:

Object / Open – Set

permit deschiderea sau modificarea obiectului, în funcție de context.

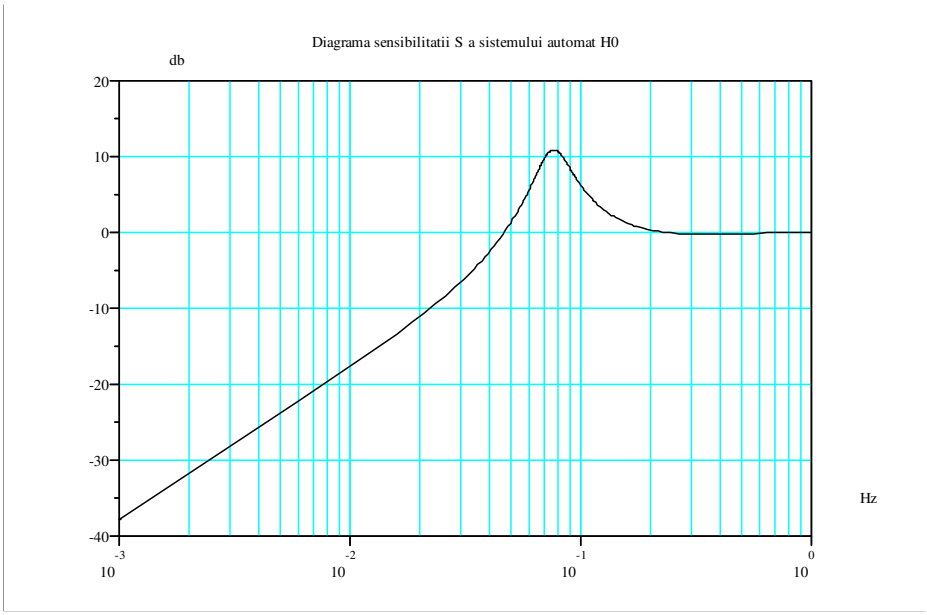


Fig. 6.29 Sensibilitatea

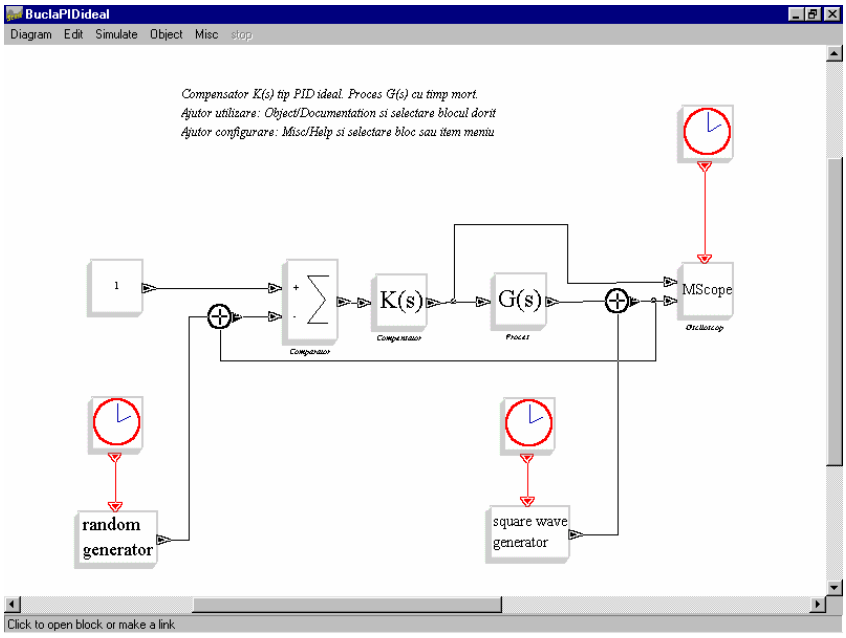

BuclaPIDideal
Diagram Edit Simulate Object Misc stop
Compensator K(s) tip PID ideal. Proces G(s) cu timp mort.
Ajutor utilizare: Object/Documentation si selectare blocul dorit
Ajutor configurare: Misc/Help si selectare bloc sau item meniu
1
K(s)
G(s)
MScope
random generator
square wave generator
Click to open block or make a link

Fig. 6.30 Fereastra Scicos cu programul BuclaPIDideal.cos.

Dacă nu se alege meniul *Object* ultima comandă se execută implicit la selectarea blocului. De exemplu, selectarea blocului ***K(s)*** din Fig. 6.30 produce

fereastra din figura următoare. Se observă că compensatorul PID ideal este neliniar, având la ieşire un bloc limitator de saturare. Selectarea acestui bloc afişează parametrii săi din Fig. 6.32.

Se deschide şi apoi se setează parametrii blocului ***G(s)*** care modelează instalaţia neautomatizată. Se introduc sub formă simbolică numărătorul şi numitorul funcţiei de transfer nominale urmărind exemplul datelor implicite din *Fig. 6.33*.

Cu datele modificate se execută comanda:

Simulate / run

Oprirea simulării se face cu comanda ***stop*** din meniul ferestrei prezentate în Fig. 6.30.

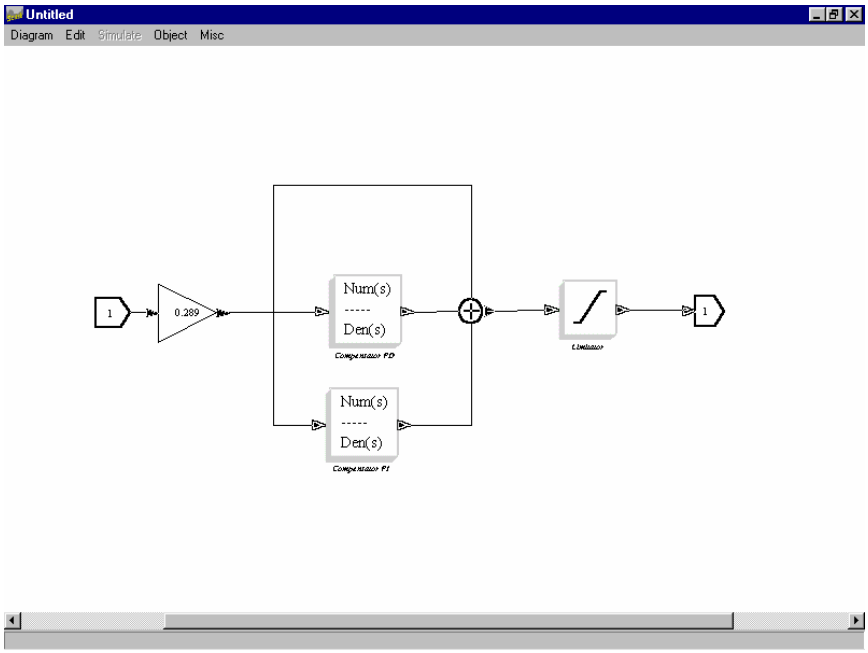


Fig. 6.31 Fereastra compensatorului K(s).

Parametrii simulării se modifică cu comanda *Simulate/Setup*. Parametrul care interesează este *Final integration time,* durata simulării.

Pentru modificarea aspectului graficului semnalelor de ieşire a sistemului simulat se selectează osciloscopul *MScope.* Parametrii graficului care ar putea fi necesar să se modifice sunt *Ymax vector*, care stabileşte valoarea maximă a ordonatei şi *Refresh period* care trebuie să fie egal cu timpul de integrare final.

Graficul din Fig. 6.34 se poate copia în documentul Word al proiectului cu comanda

File / Copy to clipboard

şi apoi ***Paste.***

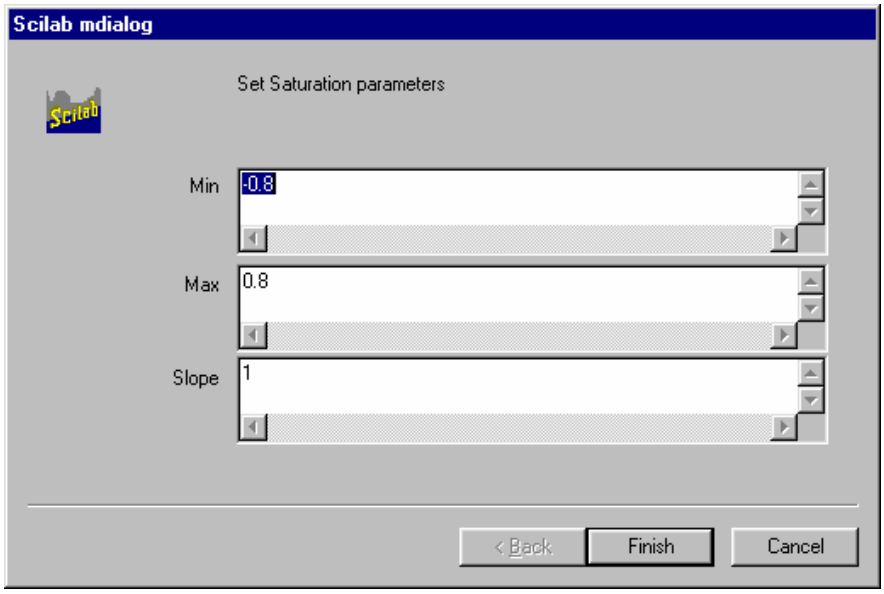


Fig. 6.32 Dialogul pentru introducerea parametrilor limitatorului semnalului de comandă.

În Fig. 6.34 se prezintă în primul grafic, cu albastru, comanda u(t) şi în al doilea grafic, cu verde, ieşirea instalaţiei automatizate. Nu apar în grafic semnalele de intrare. Referinţa este o treaptă unitară, după cum rezultă din Fig. 6.30. Perturbaţia este un semnal rectangular periodic cu amplitudinea +0,5 sau +0,5 şi perioada 30. Zgomotul este un semnal aleator cu media zero şi abaterea medie pătratică 0.05.

În momentul iniţial perturbaţia apare integral la ieşire, Fig. 6.30, care are valoarea +0,5. Compensatorul începe să compenseze perturbaţia şi să aducă ieşirea la o valoare egală cu cea a referinţei, adică unu. Dar la timpul 30 perturbaţia devine –0,5 şi ieşirea scade cu 1. Compensatorul lucrează în continuare . La timpul 60 perturbaţia îşi modifică valoarea de la +0,5 la +0,5, Deci ieşirea, la care perturbaţia se adună ca în Fig. 6.30, îşi creşte valoarea cu 1.

Este interesantă evoluţia mărimii de comandă u(t) a compensatorului. Se observă că zgomotul, deşi foarte mic, are o prezenţă semnificativă în mărimea de comandă dar nu influenţează ieşirea. Din această cauză elementul de execuţie se poate strica. Evitarea acestui efect se poate face prin reacordarea compensatorului. La momentul t=30 când a dispărut perturbaţia şi ieşire a scăzut brusc, comanda ar trebuii să fie foarte mare pentru a compensa acest efect. Din cauza limitării la +0.8 a ieşirii compensatorului, Fig. 6.32, nu se întâmplă aşa, Fig. 6.34. În schimb, la momentul 60, când perturbaţia dispare, compensatorul funcţionează normal. Aici se observă foarte bine şi efectul derivativ pronunţat care este cauza amplificării zgomotului.

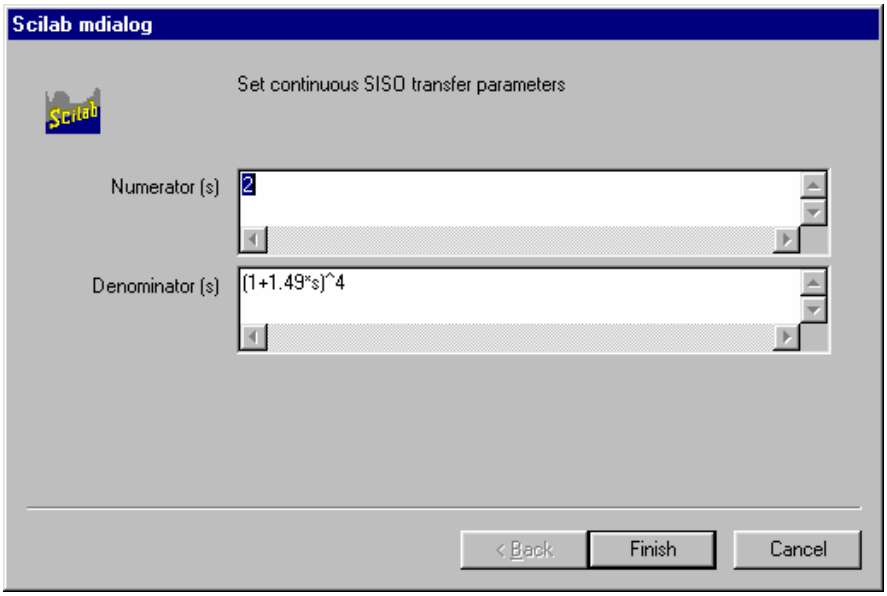


Fig. 6.33 Introducerea sub formă simbolică a funcției de transfer a instalației.

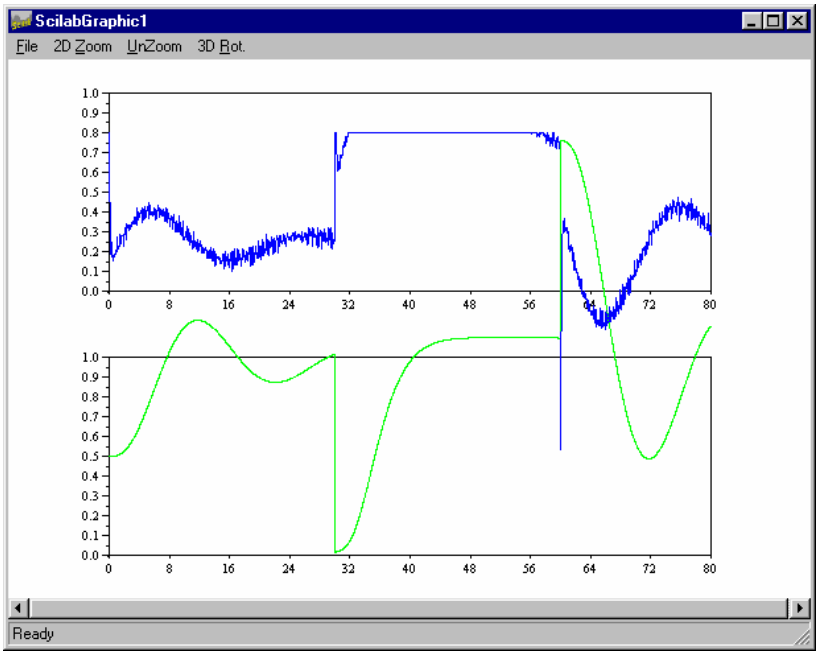


Fig. 6.34 Rezultatul simulării sistemului cu regulator numeric PID

Sistemul automat cu compensatorul acordat Ziegler – Nichols funcționează bine în regim de reglare (stabilizare) dar în regim de urmărire prezintă oscilații mari. Remediul îl constituie folosirea unui compensator cu două grade de libertate. În lipsa acestuia pornirea se face manual, treptat şi după atingerea punctului de funcționare dorit se trece în regim de reglare automată.

6.4. Problema timpului mort în proiectarea asistată de calculator

Timpul mort este un parametru foarte important al sistemelor. El are o interpretare fizică clară, caracterizând procesele de transport de material, energie sau informație. Adeseori şi procesele distribuite în spațiu sunt caracterizate aproximativ tot cu ajutorul unui timp mort. În practica curentă identificarea unui proces se face cel mai simplu cu ajutorul următorului model de tip funcție de transfer:

$$H_F(s) = \frac{Y(s)}{U(s)} = K\frac{e^{-s\tau}}{1+sT_F} \qquad (6.93)$$

în care U(s) şi Y(s) sunt intrarea şi ieşirea sistemului,

K_F : constanta de proporționalitate,

τ : constanta de timp mort,

T_F : constanta de timp.

În domeniul timp modelul (6.93) se prezintă sub forma următoarei ecuații diferențiale:

$$T_F\frac{dy(t)}{dt} + y(t) = K_F u(t-\tau) \qquad (6.94)$$

Modelul în spațiul stărilor corespunzător lui (6.94) este:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) = \left[-\frac{1}{T}\right].[x_1(t)] + \left[\frac{K}{T}\right].u(t-\tau)$$
$$y(t) = Cx(t) + Du(t) = [1].[x_1(t)] \qquad (6.95)$$

în care $x(t)=[x_1(t)]$ este vectorul stărilor iar

A,B,C şi D: matrici, parametrii modelului.

În toate cele trei modele ale sistemului timpul mort apare sub o formă necorespunzătoare. În domeniul frecvență modelul (6.93) conține timpul mort în componenta de tip exponențial:

$$H_m(s) = e^{-s\tau} \qquad (6.96)$$

iar în domeniul timp modelele (6.94) şi (6.94) depind de timpul mort în mod implicit. Eliminarea acestor dificultăți se face în mod frecvent prin două metode: aproximarea Pade şi transformarea într-un model cu timp discret.

Folosirea aproximării Pade conduce la modele de tip fracție rațională pentru funcția de transfer (6.96) care conține timpul mort. Prezentăm în continuare aproximațiile Pade de ordinul 1,2,3 şi 4.

$$e^{-s\tau} = \frac{2 - s\tau}{2 + s\tau} \tag{6.97}$$

$$e^{-s\tau} = \frac{12 - 6s\tau + s^2\tau^2}{12 + 6s\tau + s^2\tau^2} \tag{6.98}$$

$$e^{-s\tau} = \frac{120 - 60s\tau + 12s^2\tau^2 - s^3\tau^3}{120 + 60s\tau + 12s^2\tau^2 + s^3\tau^3} \tag{6.99}$$

$$e^{-s\tau} = \frac{1680 - 840s\tau + 180s^2\tau^2 - 20s^3\tau^3 + s^4\tau^4}{1680 + 840s\tau + 180s^2\tau^2 + 20s^3\tau^3 + s^4\tau^4} \tag{6.100}$$

Aproximațiile Pade de ordinul doi şi trei sunt cele mai folosite.

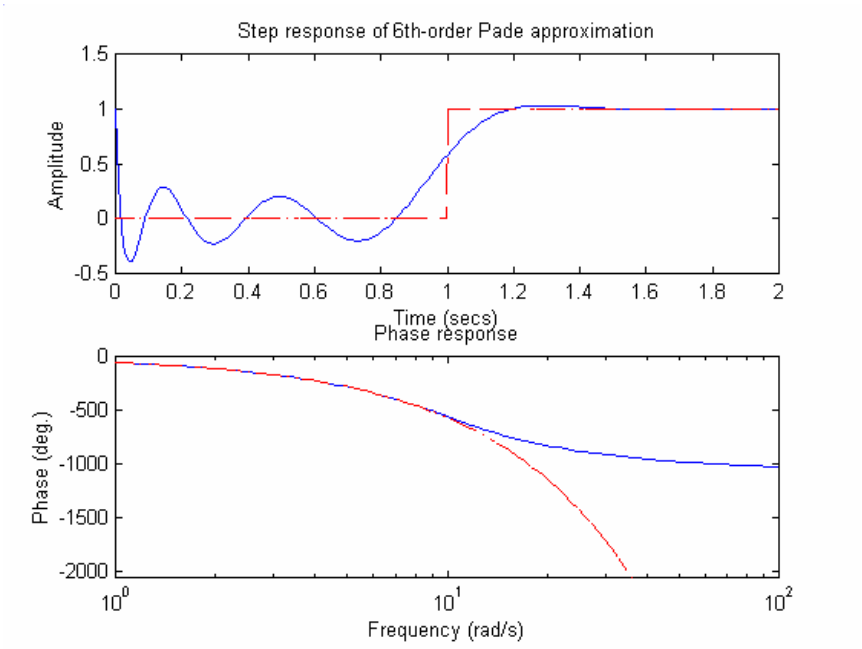


Fig. 6.35 Funcția de transfer cu timp mort aproximată studiată în domeniul timp şi domeniul frecvență.

Deoarece în practica inginerească modelul (6.93) este foarte folosit, am încercat să determinăm condiţiile in care aproximarea Pade este satisfăcătoare. Pentru aceasta am studiat în domeniul timp şi frecvenţă funcţiile de transfer de tip (6.96) înainte şi după aproximarea lor de tip Pade de ordinul 1…9. Răspunsul indicial şi diagrama Bode de fază sunt prezentate în Fig. 6.35. Modulul funcţii de transfer (6.96) este egal cu unu pentru orice frecvenţă şi nu a mai fost reprezentat grafic.

Se observă că timpul mort considerat a fost τ=1 secundă. În domeniul frecvenţă se observă că aproximarea este bună până la pulsaţia de 10 radiani/sec. În domeniul timp aproximaţia este nesatisfăcătoare datorită oscilaţiilor care apar în perioada timpului mort de o secundă. Aceste oscilaţii au aproximativ amplitudinea de 0,3 şi perioada de T_0=0,4 secunde.

Însă cel mai frecvent timpul mort apare în expresii de tipul (6.93). Aceasta înseamnă că oscilaţiile din Fig. 6.35 provocate de aproximarea Pade pot fi atenuate de către filtrul trece jos reprezentat de funcţia de transfer de ordinul unu cu care este înmulţită exponenţiala cu timp mort. Pentru o atenuare de aproximativ 100 de ori este necesar ca:

$$\frac{100}{T_F} \geq \omega_0 = \frac{2\pi}{T_0} \qquad (6.101)$$

Pentru exemplul considerat T_0=0,4 secunde şi rezultă T_F>6,3 secunde. În Fig. 6.36 se prezintă răspunsul indicial al sistemului cu funcţia de transfer (6.93) în care K=1, τ=1s şi T_F=6,3s. Se observă că oscilaţiile din perioada timpului mort Fig. 6.35 s-au micşorat de aproximativ 100 de ori.

Pentru timpi morţi diferiţi de 1 secundă perioada oscilaţiilor poate fi exprimată cu următoarea relaţia determinată prin experimente simulate:

$$T_0 = (1-0{,}1n)\tau \qquad (6.102)$$

în care n = 2…9 este ordinul aproximaţiei Pade

Din relaţiile (6.101) şi (6.102) rezultă o relaţie orientativă pentru alegerea ordinului n a aproximaţiei Pade în funcţie de raportul între timpul mort τ şi constanta de timp T_F a sistemului:

$$\frac{\tau}{T_F} = \frac{2\pi}{100(1-0{,}1n)} \qquad (6.103)$$

Intr-un mod global şi aproximativ relaţia (6.103) spune că aproximaţia Pade este utilă pentru funcţiile de transfer de tipul (6.93) dacă timpul mort nu depăşeşte 10% din constanta de timp a sistemului. Aceasta este şi limita pentru care sunt folositoare regulatoarele PID pentru conducerea unor sisteme cu această funcţie de transfer.

A doua metodă de reprezentare a timpului mort constă în folosirea modelelor cu timp discret. Pentru aceasta se introduce la intrarea blocului cu funcţia de transfer (6.93) un eşantionor şi un extrapolator de ordin zero, iar la ieşire un eşantionor, ca în Fig. 6.37.

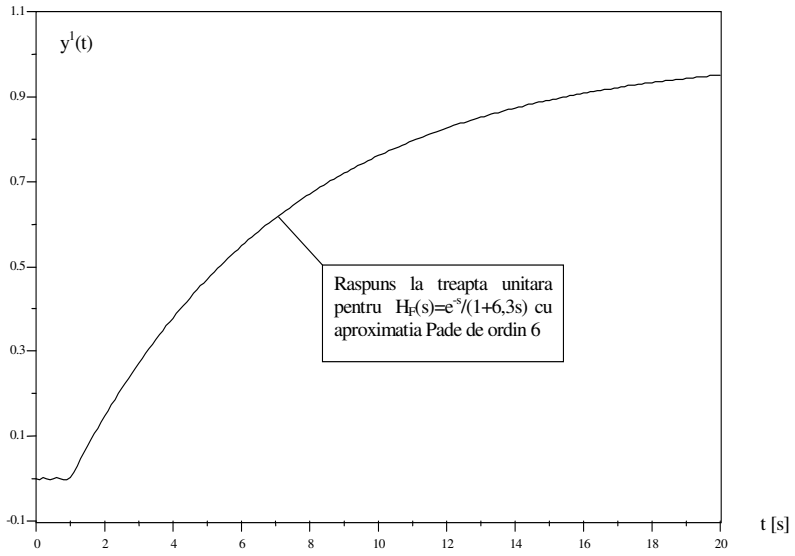


Fig. 6.36 Simularea prntru verificarea relaţiei (6.103) pentru n=6, τ=1s şi T_F=6.3s.

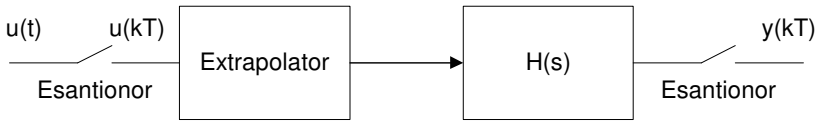


Fig. 6.37 Modelul numeric echivalent al unei funcţii de transfer.

Raportul dintre transformata z a ieşirii discretizate şi transformata z a intrării discretizate formează funcţia de transfer discretă. Pentru cazul particular (6.93) se obţine relaţia următoare:

$$H_F(z) = \frac{Y(z)}{U(z)} = K_F z^{-n} \frac{1 - e^{-T/T_F}}{z - e^{-T/T_F}} \qquad (6.104)$$

în care T este perioada de eşantionare iar n este determinat de relaţia:

$$n = \text{int}(\frac{\tau}{T}) \qquad (6.105)$$

Dacă K_F=1, T_F=1s şi T=1s atunci (6.104) devine:

$$H_F(z) = z^{-1}\frac{0.63}{z-0.37} \qquad (6.106)$$

Răspunsul la semnal treaptă unipară pentru această funcţie de transfer este prezentat în Fig. 6.38. Comparativ cu Fig. 6.36 se observă că timpul mort este reprezentat exact deşi constanta de timp a sistemului T_F este mult mai mică, 1 secundă faţă de 6,3 secunde.

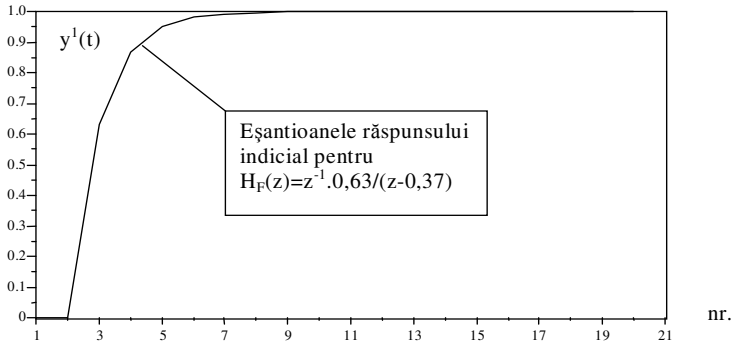


Fig. 6.38 Simularea unui model discret cu timp mort.

După cum se observă, în Fig. 6.38 valorile dintre eşantioane sunt interpolate liniar. Datorită perioadei mai de eşantionare T=1s comparativ cu constanta de timp T_F=1s răspunsul indicial are un aspect foarte aproximativ. Se poate micşora perioada de eşantionare dar în acest caz creşte ordinul modelului n determinat de relaţia (6.105).

O altă dificultate apare la folosirea modelului discret în cazul în care raportul dintre timpul mort τ şi perioada de eşantionare T în relaţia (6.105) nu este un număr întreg. Soluţia acestei probleme am găsit-o în aproximarea Pade de ordin unu (6.97) a restului.

Cele două metode de reprezentare a timpului mort le-am concretizat în două program Scilab prezentate în Fig. 6.39 şi *Fig. 6.40.*

În concluzie, pentru sistemele obişnuite, întâlnite frecvent în practică, consider că metoda aproximaţiei Pade este foarte utilă în analiza sistemelor iar metoda modelului discret în simularea sistemelor. Am folosit aproximaţia Pade de ordinul doi cu succes în analiza sistemelor automate prin metoda locului rădăcinilor. Programul de simulare KitSAS pe care l-am elaborat foloseşte metoda modelului discret fără aproximarea Pade a restului.

```
function [H]=pade(T,n)
a=[ ];b=[ ];
[nargout,nargin] = argn(0)
//   PADE  Aproximatia Pade a timpului mort.
//   [H] = PADE(T,n) determina aproximatia
//Pade de ordin n a functiei de transfer exp(-T*s).
//H este un obiect tip functie de transfer.
ni = nargin;
no = nargout;
if ni==1 then
 n = 1;
elseif n<0|T<0 then
  error('T si N trebuie sa fie nenegative.');
end
n = round(n);
// Coeficientii aproximatiei Pade
// se determina recursiv
// h[k+1] = (N-k)/(2*N-k)/(k+1) * h[k],  h[0] = 1
// exp(-T*s) ==
// Sum { h[k] (-T*s)^k } / Sum { h[k] (T*s)^k }
if T>0 then
  a = zeros(1,n+1);
  a(1) = 1
  b = zeros(1,n+1);
  b(1) = 1
for k = 1:n
    fact = T*(n-k+1)/(2*n-k+1)/k;
    %v2 = (-fact)*a(k)
    a(1,k+1) = %v2(:).';
    %v2 = fact*b(k)
    b(1,k+1) = %v2(:).';
  end
%v1 = a/b(n+1)
  a = %v1(:,$:-1:1);
  %v1 = b/b(n+1)
  b = %v1(:,$:-1:1);
else
  a = 1;
  b = 1;
end  //if T>0
for k=1:n+1
a1(k)=a(n-k+2)
b1(k)=b(n-k+2)
end  //for k
num=poly(a1, 's', 'coef')
den=poly(b1, 's', 'coef')
H=syslin('c', num)
endfunction
```

Fig. 6.39 Programul Scilab pentru construirea unei funcţii de transfer cu timp mort prin aproximarea Pade

```
function [Hz]=s2z(Hs, T, Tm)
// O functie de transfer in s
// este transformata intr-o functie de transfer in z
// Hz=s2z(Hs, T {,Tm})
// Hz - functie de transfer in z
// Hs - functie de transfer in s obtinuta cu syslin
// T  - perioada de esantionare
// Tm - timpul mort, facultativ
[nargout,nargin] = argn(0)
ni = nargin;
no = nargout;
if ni == 3 then
// restul timpului mort care nu poate fi discretizat
nmort=Tm/T;
n=int(nmort);
Tr=Tm-n*T;
// aproximatia pade2 a restului
num=poly([2, -Tr], 's', 'coef');
den=poly([2,  Tr], 's', 'coef');
pade2=syslin('c', num/den)
h=Hs*pade2;
ssd=dscr(h, T);
sd=ss2tf(ssd);
Hmort=1/(%z^n);
Hz=Hmort*sd;
else
ssd=dscr(Hs, T);
Hz=ss2tf(ssd);
end
endfunction
```

Fig. 6.40 Programul Scilab pentru transformarea unui model continuu într-un model discret cu timp mort

6.5. Sisteme automate cu regulatoare cu model intern

Majoritatea regulatoarelor automate folosite în instalaţiile pentru construcţii sunt clasice, de tipul P, 2P, 3P, PI, PD sau PID[152]. Alte regulatoare care mai pot fi folosite sunt de tipurile: adaptiv, neural, fuzzy, neliniar, cu modern intern, etc. Regulatoarele care conţin explicit un model intern[153] al instalaţiei automatizate, apărute în ultimii zece ani, au o şansă mare să cunoască o utilizare mai largă deoarece sunt relativ simple, permit eliminarea totală a perturbaţiei şi urmărirea exactă a referinţei şi pot fi implementate cu ajutorul regulatoarelor clasice de tip PID. Folosirea lor permite înţelegerea mai uşoară a folosirii regulatoarelor cu două grade de libertate, a robusteţii stabilităţii şi performanţelor sistemului, consideră eroarea de modelare şi permite evitarea saturării comenzii

[152] Proporţional, bipoziţional, tripoziţional, Proporţional-Integrativ, Proporţional-Derivativ sau Proporţional-Integrativ-Derivativ.

[153]Conducere pe bază de model intern - Internal Model Control - IMC

fără măsuri speciale antisaturare. Aceste regulatoare pot fi proiectate simbolic fără ajutorul unor metode grafice cum este metoda locului rădăcinilor şi din această cauză proiectarea lor este mai intuitivă şi are o valoare educaţională mai mare. Alături de feedforward şi reglarea în cascadă, reglarea pe bază de model intern reprezintă o arhitectură alternativă la bucla clasică de reglare.

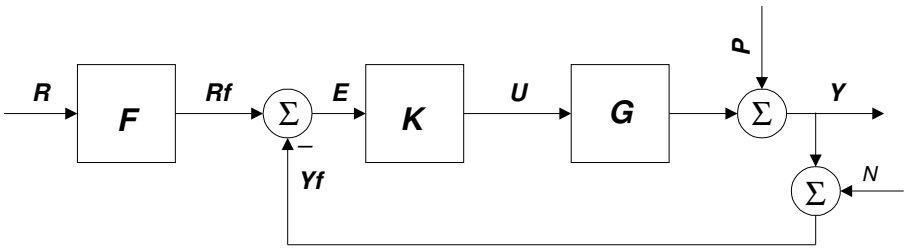


Fig. 6.41 Buclă cu regulator clasic cu două grade de libertate.

6.5.1. Definirea regulatorului cu model intern

Bucla de reglare clasică din este echivalentă cu bucla cu regulator cu model intern din Fig. 6.42 dacă este respectată următoarea relaţie dintre compensatorul clasic K şi compensatorul Q al buclei cu model intern:

$$K = \frac{Q}{1 - G_m Q} \tag{6.107}$$

sau

$$Q = \frac{K}{1 + G_m K} \tag{6.108}$$

în care G_m este modelul instalaţiei automatizate G.

În practică se proiectează compensatorul Q şi apoi se determină cu ajutorul relaţiei (6.107) compensatorul clasic K. Dacă nu rezultă un compensator PID se face o aproximare.

Să vedem cum se proiectează compensatorul Q al regulatorului cu model intern G_m.

Relaţia dintre intrările R (referinţa), P (perturbaţia) şi N (zgomotul) şi ieşirea Y este:

$$Y = \frac{GQ}{1 + Q(G - G_m)} FR + \frac{1 - G_m Q}{1 + Q(G - G_m)} P - \frac{GQ}{1 + Q(G - G_m)} N \tag{6.109}$$

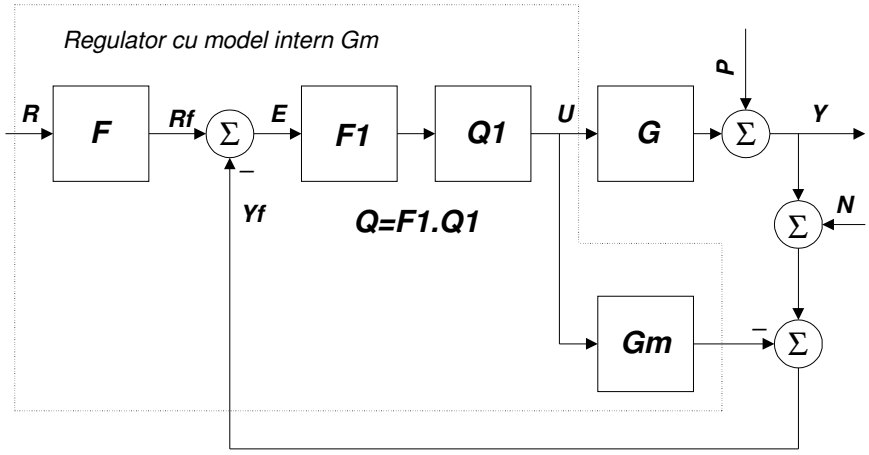


Fig. 6.42 Buclă cu regulator cu două grade de libertate şi model intern explicit G_m.

6.5.2. Proiectarea regulatorului cu model intern

Pentru proiectarea compensatorului Q se iau în considerare două cazuri: a) modelul G_m este identic cu procesul G şi b)modelul G_m este diferit de procesul G. Să le examinăm pe rând.

a) Modelul G_m este identic cu procesul G.

În această situaţie există relaţia:

$$G_m = G \tag{6.110}$$

Cu această condiţie (6.109) devine:

$$Y = G_m QFR + (1 - G_m Q)P - G_m QN \tag{6.111}$$

Comanda U şi eroarea E vor avea următoarele expresii:

$$U = QFR - QP - QN \tag{6.112}$$

$$E = (1 - G_m Q)FR - (1 - G_m Q)P - (1 - G_m Q)N \tag{6.113}$$

Dacă este compensatorul Q este determinat cu relaţia:

$$Q = \frac{1}{G_m} \tag{6.114}$$

atunci perturbația P este eliminată în (6.111) iar ieşirea Y sistemului automat depinde numai de R şi N:

$$Y = FR - N \tag{6.115}$$

Dacă nu există zgomot, N=0, atunci se poate alege prefiltrul F astfel încât ieşirea Y să urmărească în modul dorit referința R. Din Fig. 6.42 se observă că daca modelul este perfect şi (6.110) este îndeplinită atunci sistemul este în buclă deschisă în regim de urmărire. Cu alte cuvinte reacția apare numai dacă perturbația P este diferită de zero. Deci singurul element care afectează urmărirea este prefiltrul F.

Pentru un model G_m perfect sistemul automat funcționează ideal în regim de urmărire a referinței R şi în regim de înlăturare a perturbației P dacă compensatorul Q este o funcție de modelul G_m conform relației (6.114).

b) Modelul G_m este diferit de procesul G.

În acest caz atât perturbația P cât şi eroarea de modelare sunt transmise prin reacția negativă la intrare ca în Fig. 6.42 şi compensatorul Q trebuie să fie dezacordat pentru a face față noii situații. Relația (6.114) nu mai este valabilă. Din păcate proiectarea compensatorului Q chiar în cazul a) al modelelor prefecte cu ajutorul relației (6.114) nu este posibilă, în practică, deoarece conduce la un compensator Q care poate să fie instabil, sau nerealizabil[154] fizic, sau neadecvat, sau să nu conducă la performanțele dorite[155] pentru Y, E şi U. Evitarea acestor situații se face în patru etape.

*Etapa1.*În primul rând transformăm compensatorul Q astfel încât să fie format din două blocuri Q_1 şi F_1 ca în Fig. 6.42. Noul compensator va fi:

$$Q = F_1 Q_1 \tag{6.116}$$

Etapa 2. Se determină Q_1 în funcție de modelul procesului G_m astfel încât acesta să fie *stabil şi realizabil fizic*. O variantă[156] o constituie folosirea următoarei relații:

$$Q_1 = \frac{1}{G_{m1}} \tag{6.117}$$

în care G_{m1} este un model al procesului automatizat care conține toate componentele de fază minimă ale lui G_m, adică polii şi zerourile din semiplanul stâng.

Datorită modului în care a fost ales, G_{m1} este invertibil iar Q_1 calculat cu (6.117) este stabil şi realizabil fizic.

Etapa3. In această etapă impunem condiția ca Q să fie *adecvat* şi să satisfacă *performanțele dorite* prin introducerea unei filtrări suplimentare a erorii E cu ajutorul filtrului F_1 din Fig. 6.42. Pentru un răspuns aperiodic al sistemului automat se alege F_1 astfel:

---

[154] Nu respectă principiul cauzalității care spune că efectul trebuie să nu anticipeze cauza.

[155] De exemplu nu înlătură zgomotul N sau nu este suficient de rapid.

[156] Există şi alte posibilități.

$$F_1 = \frac{1}{(\lambda s + 1)^n} \tag{6.118}$$

în care λ este un parametru de tip constantă de timp cu care se pot asigura anumite performanțe cum ar fi durata procesului tranzitoriu sau banda de trecere. Cu cât λ este mai mare cu atât este mai mare şi robustețea sistemului automat.

Ordinul n al filtrului F_1 este ales suficient de mare astfel încât compensatorul Q determinat de (6.117) să fie *adecvat*, adică ordinul numitorului lui Q să fie mai mare sau cel mult egal cu ordinul numărătorului lui Q.

Etapa 4. Înlocuind (6.117) şi (6.118) în (6.116) se obține cu ajutorul relației (6.107) expresia analitică a compensatorului clasic dorit:

$$K = \frac{Q}{1 - G_m Q} = \frac{1}{s} \frac{G_{m1}^{-1}}{\left((\lambda s + 1)^n - G_{m2}\right)/ s} = \frac{1}{s} f \tag{6.119}$$

în care $G=G_{m1}.G_{m2}$ iar G_{m1} este partea de fază minimă a modelului iar G_{m2} respectiv partea de fază neminimă a modelului procesului, adică partea care conține zerourile din semiplanul drept şi întârzierile cu timp mort.

Dezvoltând în (6.119) pe *f(s)* în serie Taylor se obține prin aproximare un regulator PID

$$K = \frac{1}{s}\left(f(0) + f'(0)s + \frac{f''(0)}{2} s^2 + \cdots \right) \cong K_r (1 + \frac{1}{T_i s} + T_d s) \tag{6.120}$$

cu acordarea următoare:

$$K_r = f'(0) \tag{6.121}$$

$$T_i = \frac{f'(0)}{f(0)} \tag{6.122}$$

$$T_d = \frac{f''(0)}{2 f'(0)} \tag{6.123}$$

6.5.3. Exemplu de proiectare

Să considerăm un model de tip Strejc pentru instalația care urmează să fie automatizată şi care are forma următoare:

$$G_m = \frac{K_p e^{-\tau s}}{(Ts + 1)^n} \tag{6.124}$$

Partea de fază minimă a modelului procesului G_m care conține numai polii şi zerourile din semiplanul stâng este:

$$G_{m1} = \frac{K_p}{(Ts+1)^n} \tag{6.125}$$

Partea Q_1 a compensatorului Q din Fig. 6.42 care este o funcție de modelul intern al procesului se calculează în acest caz cu (6.117) rezultând:

$$Q_1 = \frac{1}{G_{m1}} = \frac{(Ts+1)^n}{K_p} \tag{6.126}$$

Ordinul numărătorului lui Q_1 este mai mare cu doi decât ordinul numitorului, deci Q_1 este *neadecvat*. Ordinul n al filtrului F_1 ar trebuii să fie ales egal cu doi sau mai mare pentru ca Q determinat de (6.116) să fie *adecvat*. Vom alege totuşi în (6.118) n=1 pentru obținerea unui regulator mai simplu iar problema adecvării o rezolvăm mai târziu. Deci,

$$F_1 = \frac{1}{(\lambda s+1)} \tag{6.127}$$

Compensatorul Q din Fig. 6.42 va fi conform relației (6.116) următorul:

$$Q = \frac{(Ts+1)^n}{K_p(\lambda s+1)} \tag{6.128}$$

Cu relația (6.107) se obține compensatorul clasic K al sistemului automat din .

$$K = \frac{Q}{1-G_m Q} = \frac{1}{K_p}\frac{(Ts+1)^n}{\lambda s+1-e^{-\tau s}} \tag{6.129}$$

Acest compensator este un predictor Smith. Se poate face următoarea aproximare[157]:

$$e^{-\tau s} \approx 1-\tau s \tag{6.130}$$

şi (6.129) devine:

$$K = \frac{1}{K_p}\frac{(Ts+1)^n}{(\lambda+\tau)s} = \frac{1}{K_p(\lambda+\tau)}\frac{1}{s}f \tag{6.131}$$

în care

$$f = (Ts+1)^n \tag{6.132}$$

[157] Există şi aproximări mai bune. În mod curent se folosesc aproximări Pade de ordinul unu sau doi.

poate fi dezvoltată în serie Taylor

$$f = (Ts+1)^n \cong 1 + nTs + \frac{1}{2}nT^2(n-1)s^2 \qquad (6.133)$$

Înlocuind pe f în (6.131) se obține:

$$K = \frac{nT}{K_p(\lambda+\tau)}\left[1 + \frac{1}{nTs} + (n-1)\frac{T}{2}s\right] \qquad (6.134)$$

care este un regulator PID ideal de forma

$$K = K_r(1 + \frac{1}{T_i s} + T_d s) \qquad (6.135)$$

cu acordarea

$$K_r = \frac{nT}{K_p(\lambda+\tau)} \qquad (6.136)$$

$$T_i = nT \qquad (6.137)$$

$$T_d = (n-1)\frac{T}{2} \qquad (6.138)$$

Compensatorul ideal PID (6.135) este *neadecvat* deoarece are un derivator ideal şi ordinul numărătorului este mai mare decât ordinul numitorului. La acest rezultat s-a ajuns deoarece nu am ales în (6.127) pentru F_1 forma adecvată. Acest neajuns se înlătură în practică alegând un compensator PID ideal cu filtrarea componentei derivative astfel

$$K = K_r(1 + \frac{1}{T_i s} + \frac{T_d s}{0{,}2T_d s + 1}) \qquad (6.139)$$

Din (6.111), (6.112), (6.124) şi (6.128) rezultă ieşirea şi comanda sistemului în regim de urmărire:

$$Y = QG_m FR = \frac{e^{-\tau s}}{\lambda s + 1} FR \qquad (6.140)$$

$$U = QFR = \frac{(Ts+1)^n}{K_p(\lambda s+1)} FR \qquad (6.141)$$

Dacă regulatorul are un singur grad de libertate, deci prefiltrul F=1, din relațiile precedente se constată că parametrul λ determină performanțele sistemului automat. Pentru a nu avea comenzi U prea energice rezultă din (6.140) şi (6.141) că este necesar ca banda de trecere a sistemului automat să fie mai mică sau egală cu banda de trecere a procesului. Astfel un compromis[158] rezonabil a valorii lui λ în (6.136) din punct de vedere a robusteții stabilității şi robusteții performanțelor este:

$$\lambda \geq nT \qquad (6.142)$$

Pentru a ilustra cât de bună este acordarea propusă pentru modelul Strejc să considerăm procesul cu modelul particular

$$G_m = \frac{7e^{-0.6s}}{(3s+1)^4} \qquad (6.143)$$

Parametrii propuşi pentru regulator (6.136), (6.137), şi (6.138) sunt:

$$\begin{aligned} K_r &= 0{,}136 \\ T_i &= 12\,[s] \\ T_d &= 4{,}5\,[s] \end{aligned} \qquad (6.144)$$

Regulatorul PID (6.139) cu filtrarea acțiunii derivative va fi atunci

$$K(s) = K_r\left(1+\frac{1}{T_i s}+\frac{T_d s}{0{,}2T_d s+1}\right) = 0{,}136\left(1+\frac{1}{12s}+\frac{4{,}5s}{0{,}9s+1}\right) \qquad (6.145)$$

[158] Condițiile de robustețe a stabilității şi de robustețe a performanțelor sunt contradictorii.

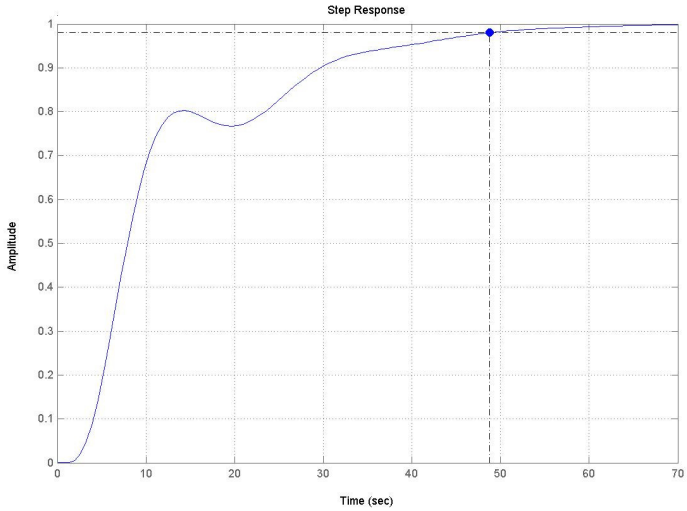


Fig. 6.43 Răspunsul indicial al sistemului automat cu procesul (6.143) şi regulatorul (6.145).

În privința performanțelor sistemului automat se constată din răspunsul indicial prezentat în Fig. 6.43 că durata regimului tranzitoriu este de 48,8 secunde, adică cea impusă prin condiția (6.142). Marginea de amplificare GM din Fig. 6.44 pentru sistemul automat este 3,28 , marginea de fază PM este 89,6 grade, marginea de întârziere $\Delta\tau/\tau$ este 18,8 iar valoarea de vârf a sensibilității M_s este 3,82 dB. Din Fig. 6.44 se observă că marginea de fază nu este semnificativă, dar restul valorilor indică o robustețe a stabilității foarte bună. Dacă se dublează valoarea constantei de proporționalitate a regulatorului din (6.144) K_r=0,0226 se obțin următoarele performanțe: marginea de amplificare GM=1,63 , marginea de fază PM=41,9 grade, marginea de întârziere $\Delta\tau/\tau$ =2,54, suprareglarea 20% şi durata procesului tranzitoriu 62,1 secunde. Sistemul nu mai este robust. Mulți autori consideră regulatoarele cu model intern o alternativă viabilă la regulatoarele clasice. După părerea mea aceste regulatoare sunt avantajoase atunci când se încearcă rezolvarea *simbolică* a problemei proiectării compensatoarelor pentru modele ale proceselor diferite de cele clasice, de exemplu Modelul Kupfmuller sau modelul Strejk. Echivalarea posibilă a acestor compensatoare cu compensatoarele clasice PID permite proiectarea inginerească a sistemelor automate pe baze mai intuitive şi prin luarea în considerare a robusteții stabilității şi performanțelor si a valorilor mari pentru semnalul de comandă U.

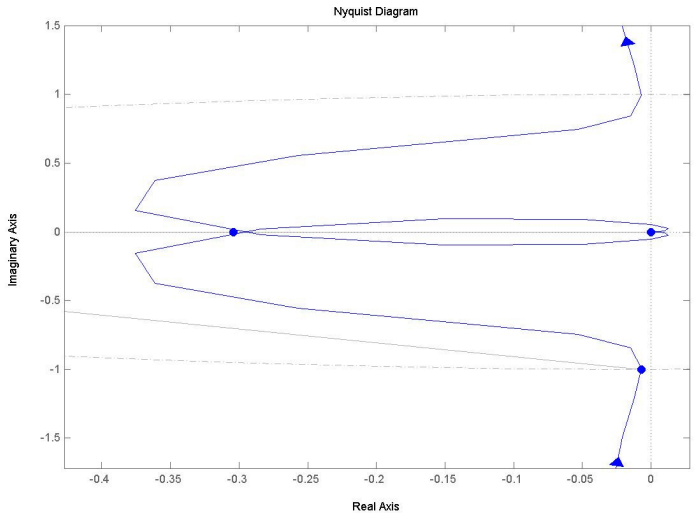


Fig. 6.44 Hodograful Nyquist pentru sistemul automat cu bucla deschisă având procesul (6.143), şi regulatorul (6.145).

6.6. Sisteme adaptive

Sistemele automate bine acordate au calitatea remarcabilă de a fi insensibile la modificări mic ai parametrilor proceselor conduse. Dacă aceşti parametrii se modifică este necesară şi adaptarea regulatoarelor la noua situaţie.

Sistemele adaptive se deosebesc de sistemele automate clasice prin apariţia unei noi bucle ca în Fig. 6.45 sau *Fig. 6.47*.

6.6.1. Sisteme adaptive cu autoacordare

Regulatoarele cu autoacordare moderne sunt microcalculatoare care au trei blocuri ca în Fig. 6.45: identificatorul lui *F(s),* sintetizatorul lui *C(s)* şi compensatorul *C(s)*

Există multe metode de identificare a funcţiei de transfer *F(s)* a procesului condus din instalaţie. Unele dintre acestea, mai simple, au fost algoritmizate şi folosite la autoacordare sau acordarea automată[159] a regulatoarelor.

Identificarea pentru acordarea automată se face presupunând că modelul procesului este de tip ARMAX (Auto – Regresive Moving Average with eXogenous input).

[159] Autoacordarea se face la cerere iar acordarea automată are loc în tot timpul procesului de reglare automată.

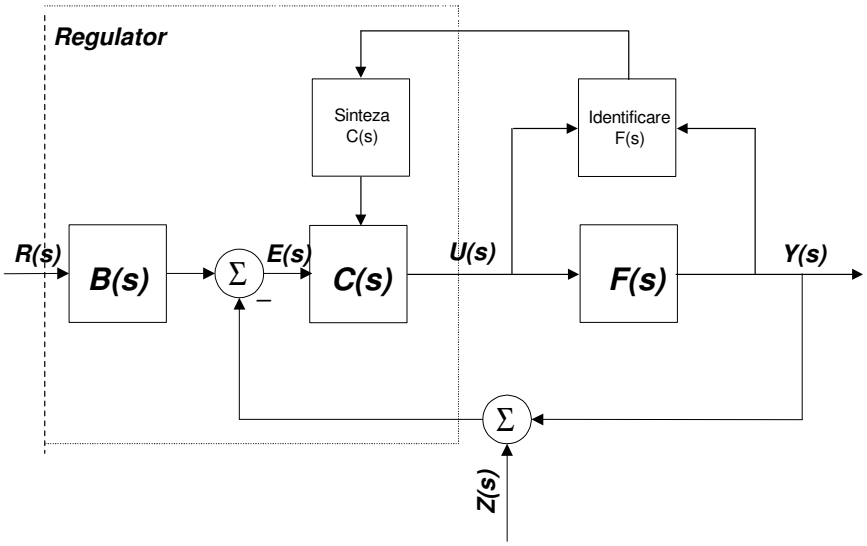


Fig. 6.45 Structura unui sistem cu autoacordare

6.6.2. Sisteme adaptive cu model de referință

Schema bloc a unui sistem automat modern, cu reacție de la perturbație, folosit în construcții este prezentată în *Fig. 6.46*. Relația dintre ieşirea sa Y şi cele trei intrări, referința R, perturbația P şi zgomotul Z, este următoarea:

$$Y = \frac{BCF}{1+CF}R + \frac{D-FA}{1+CF}P - \frac{CF}{1+CF}Z \qquad (6.146)$$

în care A(s) este compensatorul perturbației,

B(s) : prefiltrul referinței,

C(s) : compensatorul erorii,

D(s) : funcția de transfer perturbație – clădire,

F(s) : funcția de transfer a părții fixe a instalației, incluzând elementul de execuție, procesul şi traductorul.

Se observă că dacă compensatorul perturbației are funcția sa de transfer *A=D/F* atunci influența perturbației este nulă. Din păcate funcțiile de transfer *F* şi în special *D* nu sunt cunoscute şi compensatorul *A* nu poate fi determinat cu precizie. Mai mult, ele se modifică în timp. Tot din relația (6.146) se constată însă, că dacă compensatorul erorii există şi are o funcție de transfer C destul de mare, efectul perturbației este dramatic scăzut chiar în prezența unui risc de necunoaştere sau modificare a proceselor conduse. Creşterea lui *C* poate provoca însă pierderea stabilității.

Pentru înlăturarea efectelor modificării procesului condus mi-am propus să studiez un sistem automat cu regulator cu model de referință. Schema bloc a unui astfel de sistem aplicat la situația existentă în instalațiile pentru construcții este prezentată în *Fig. 6.47*. A apărut un singur bloc nou *G,* care reprezintă modelul dorit pentru funcția de transfer a sistemului automat. De data aceasta compensatorul *C* tinde să anuleze eroarea de modelare între ieşirea sistemului automat şi ieşirea modelului. Ieşirea sistemului automat este:

$$Y = \frac{F(B+CG)}{1+CF}R + \frac{D-FA}{1+CF}P - \frac{CF}{1+CF}Z \qquad (6.147)$$

Comparând relațiile (6.146) şi (6.147) observăm că sistemul cu regulator cu model de referință din *Fig. 6.47* se comportă la fel cu sistemul modern din *Fig. 6.46* în regim de reglare şi în regim de filtrare. Se modifică comportarea în regim de urmărire. Dacă *C* este un compensator proporțional cu constanta de proporționalitate *K* foarte mare atunci perturbația P este înlăturată iar ieşirea sistemului automat va fi:

$$Y = GR + FZ \qquad (6.148)$$

Funcția de transfer a sistemului automat în regim de urmărire va fi *G*, aşa cum ne-am dorit, iar funcția de transfer în regim de filtrare va fi *F*.

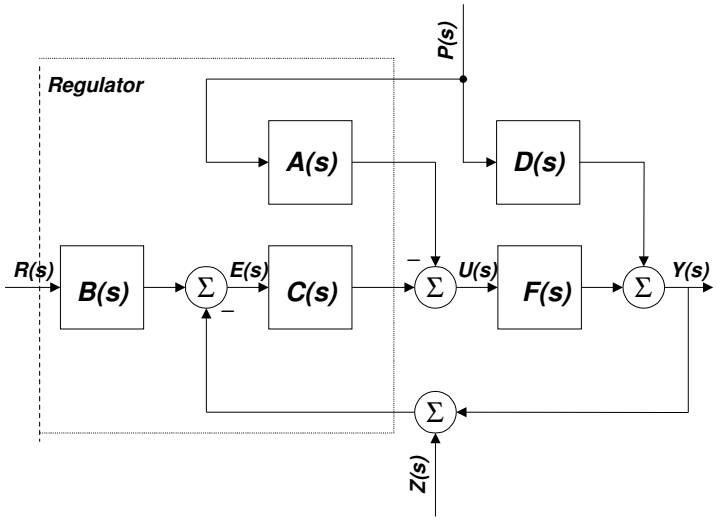


Fig. 6.46 Schema bloc tipică a unui sistem automat modern

Din păcate în practică *K* nu poate fi foarte mare deoarece sistemul devine instabil. O soluție constă în folosirea unui regulator industrial PID. În lucrare ne-am propus să proiectăm un regulator cu model de referință care foloseşte un

compensator al erorii de modelare tip PID şi să studiem comportarea sa comparativ cu cazul ideal descris de relația (6.148).

Considerăm un sistem automat de reglare a temperaturii într-o clădire. Din relația (6.147) rezultă că modelul de referință influențează doar comportarea sistemului în regim de urmărire. Pentru simplificarea analizei nu mai studiem efectul reacției de la perturbație ți efectul zgomotului. Deci *A=0*, *D=1* şi *Z=0*.

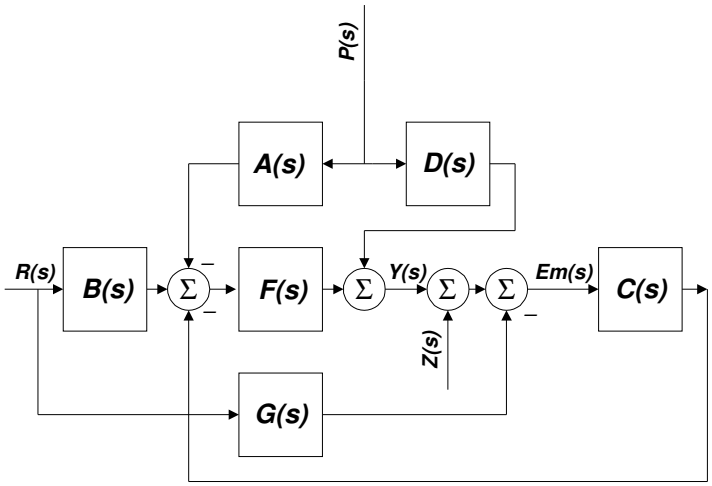


Fig. 6.47 Sistem automat modern cu regulator autoacordabil

Relația (6.147) devine:

$$Y = \frac{F(B+CG)}{1+CF}R + \frac{1}{1+CF}P \qquad (6.149)$$

În felul acesta se va studia în regim de urmărire influența regulatorului cu model de referință iar în regim de reglare influența compensatorului PID asupra stabilității.

Pentru partea fixă a instalației se acceptă funcția de transfer:

$$F(s) = \frac{1}{(1+Ts)^4} = \frac{1}{(1+s)^4} \qquad (6.150)$$

Dacă toate cele patru constante de timp *T* sunt în această relație egale cu 1 oră rezultă prin simulare că durata procesului tranzitoriu este de 10 ore. Constanta de proporționalitate a părții fixe în (6.150) este K_f=1 [V/V].

Instalația poate să urmărească modelul impus în condiții bune numai dacă este mai rapidă decât modelul. Pentru realizarea acestei condiții alegem un model cu o durată a procesului tranzitoriu de zece ori mai mare decât a instalației (6.149), adică aproximativ 100 ore. Funcția sa de transfer va fi:

$$G(s) = \frac{0{,}03}{s + 0{,}03} \qquad (6.151)$$

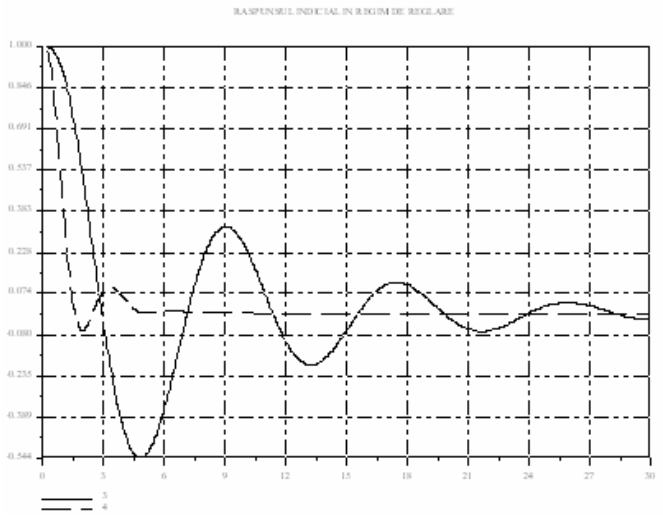


Fig. 6.48. Simularea sistemului automat cu regulator cu model de referință. Regimul de reglare. Curba 3 corespunde lui F(s) determinat de (4) cu T=1 h iar curba 4 corespunde lui F(s) cu T=0,5 h.

Proiectăm compensatorul PID al erorii de modelare prin metoda locului rădăcinilor pentru o suprareglare de 50% recomandabilă în regim de reglare. Dacă zerourile compensatorului sunt identice cu polii părții fixe F(s) pentru T=1 h rezultă constanta de proporționalitate în buclă deschisă k_0=1 şi următoarea funcție de transfer pentru compensatorul PID al erorii de modelare:

$$C(s) = \frac{(s+1)(s+1)}{s} = 2(1 + \frac{1}{2s} + 0{,}5s) \qquad (6.152)$$

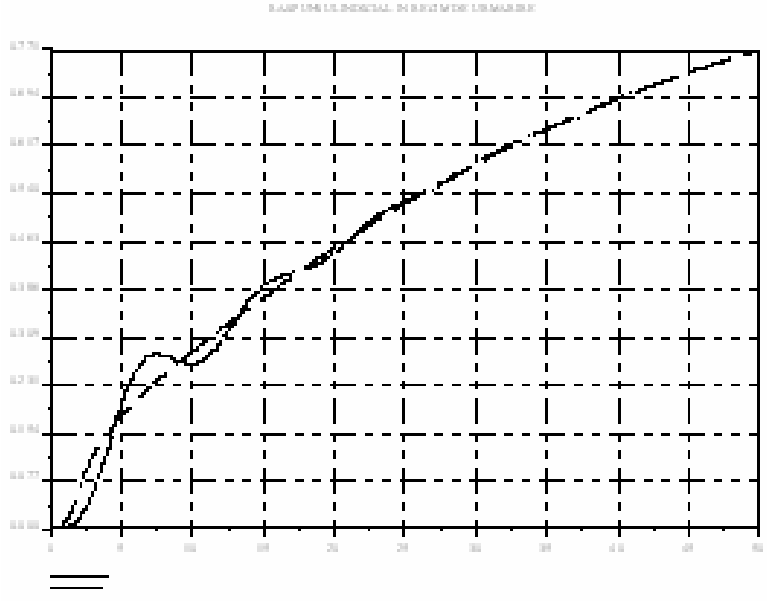


Fig. 6.49. Simularea sistemului automat cu regulator cu model de referință. Regimul de urmărire. Curba 1 corespunde lui F(s) determinat de (4) cu T=1 h iar curba 2 corespunde lui F(s) cu T=0,5 h. Modelul de referință este lent, relația 6.

Am elaborat un program de simulare al sistemului automat cu regulator cu model de referință în limbajul Scilab (lista 1). Răspunsul în regim de urmărire este prezentat în Fig. 6.48. Se observă că este îndeplinită condiția de suprareglare de 50%. Perioada oscilațiilor este de aproximativ T_0=10 h. Pentru a împiedica apariția oscilațiilor şi în regim de urmărire se introduce un prefiltru trece jos cu banda de trecere egală cu $1/T_B$=0,01. Funcția sa de transfer va fi:

$$B(s) = \frac{1}{1 + T_B s} = \frac{0{,}01}{s + 0{,}01} \qquad (6.153)$$

Simularea în regim de urmărire este prezentată în Fig. 6.49. Curba 1, corespunzătoare instalației cu funcția de transfer 5, urmăreşte, cu unele oscilații la început, modelul 6.

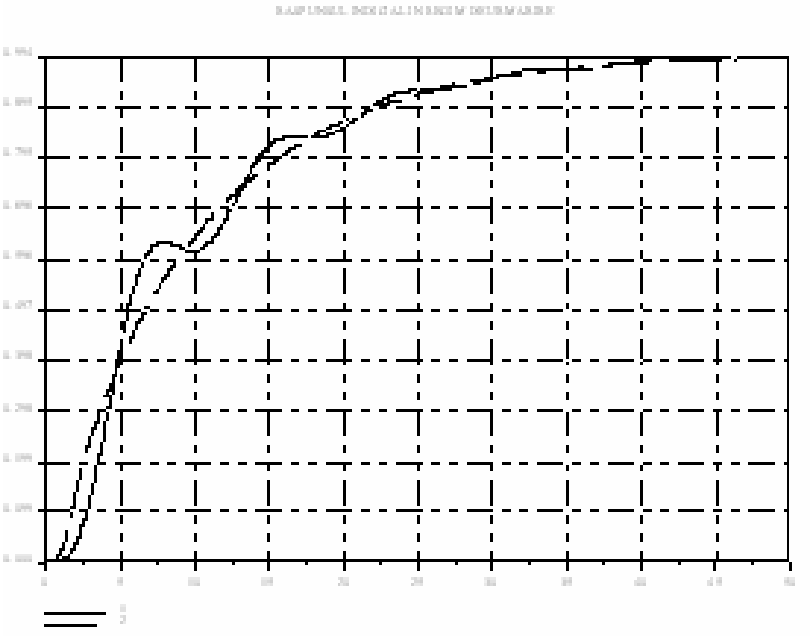


Fig. 6.50 Simularea sistemului automat cu regulator cu model de referință. Regimul de urmărire. Curba 1 corespunde lui F(s) determinat de (4) cu T=1 h iar curba 2 corespunde lui F(s) cu T=0,5 h. Modelul de referință este rapid

Deoarece sistemul proiectat are regulatorul adaptiv în regim de urmărire ne aşteptăm ca o schimbare semnificativă a funcției de transfer a părții fixe să nu

modifice mult curba 1 din Fig. 6.49. Să considerăm că partea fixă are următoarea funcție de transfer:

$$F(s) = \frac{16}{(s+2)^4} \qquad (6.154)$$

Comparând relațiile (6.150) cu (6.154) se constată o modificare cu 100% a constantelor de timp T, de la 1 h la 0,5 h. Simularea instalației (6.154) cu regulator cu model de referință în regim de urmărire conduce la răspunsul indicial prezentat sub forma curbei 2 din Fig. 6.49. Diferențele față de curba 1 sunt minore. În schimb pentru regimul de reglare din Fig. 6.48 sistemul automat răspunde foarte diferit în cazul celor două instalații, caracterizate de funcțiile de transfer (6.150) şi (6.154).

Lista 1. Programul Scilab pentru simularea sistemului automat.

```
// Sistem automat cu regulator auoacordabil cu model de referinta
s=poly(0, 's');
pB=0.1/(s+0.1);  // Prefiltrul de urmarire
pC=1*(s+1)*(s+1)/s;  // Compensatorul PID al erorii de modelare
pG=0.03/(s+0.03);  // Modelul sistemului automat in regim de urmarire

// Regim de urmarire cu T=1
pF=1/((s+1)^4);   // Partea fixa a instalatiei
pHu=pF*(pB+pC*pG)/(1+pC*pF);
tmax=50;  timp=[0:tmax/2000:tmax];
Hu=syslin('c', pHu); Y1=csim('step', timp, Hu);

// Regim de reglare cu T=1
pHr=1/(1+pC*pF);
tmax=30;  timp=[0:tmax/2000:tmax];
Hr=syslin('c', pHr); Y1r=csim('step', timp, Hr);

// Regim de urmarire cu T=0.5
pF=16/((s+2)^4); // Parte fixa a instalatiei
pHu=pF*(pB+pC*pG)/(1+pC*pF);
tmax=50;  timp=[0:tmax/2000:tmax];
Hu=syslin('c', pHu); Y2=csim('step', timp, Hu);
xbasc(0);  xset("window", 0);  xselect();  xset("use color", 0);
plot2d([timp',timp'], [Y1', Y2'], [1,2], "121", "1@2");
xgrid(5);       xtitle('RASPUNSUL INDICIAL IN REGIM DE
URMARIRE');

// Regim de reglare cu T=0.5
pHr=1/(1+pC*pF);
tmax=30;  timp=[0:tmax/2000:tmax];
Hr=syslin('c', pHr); Y2r=csim('step', timp, Hr);
xbasc(1);  xset("window", 1);  xselect();  xset("use color", 0);
plot2d([timp',timp'], [Y1r', Y2r'], [1,2], "121", "3@4");
xgrid(5);       xtitle('RASPUNSUL INDICIAL IN REGIM DE
REGLARE');
```

În sfârşit, este interesant să urmărim cum funcționează sistemul automat dacă modelul de referință al regulatorului este mai rapid. În locul funcției de transfer (6.151) să adoptăm următorul model:

$$G(s) = \frac{0{,}1}{s + 0{,}1} \qquad (6.155)$$

Simularea în regim de urmărire cu noul model este prezentată în Fig. 6.50. Comparând cu Fig. 6.49 se constată că proprietăţile de autoacordare se păstrează, doar oscilaţiile datorate regimului tranzitoriu sunt mai pronunţate.

În concluzie, regulatorul cu model de referinţă, două grade de libertate şi compensator PID al erorii de modelare poate fi folosit cu succes în regim de *urmărire* pentru conducerea proceselor din instalaţiile pentru construcţii într-o structură de forma celei din *Fig. 6.47*. In regim de *reglare* s-au *filtrare* acest regulator nu este recomandabil. Regulatorul cu model de referinţă constituie o alternativă la regulatorul autoacordabil. În acest caz regulatorul este actualizat fără o operaţie intermediară de identificare. Oscilaţiile amortizate care apar în răspunsul sistemului automat cu regulator cu model de referinţă pot produce dificultăţi în practică prin excitarea unor moduri proprii nemodelate ale instalaţiei. Pentru înlăturarea acestui efect este necesară o frecvenţă de eşantionare suficient de mare.

7. Sisteme cu eşantionare

Pentru multe sisteme se folosesc *modele discrete cu eşantionare*. În această categorie intră sistemele organizaţionale, economice, financiare, sociale sau sistemele tehnice cu calculatoare. Modelele prezentate în continuare pentru acest tip de sisteme au o arie largă de aplicare deoarece presupun prezenţa la intrarea în sistem a unui semnal determinist combinat cu un semnal aleator. In felul acesta modelele pot fi folosite nu numai la studiul sistemelor dar şi la studiul semnalelor. Pentru aceasta semnalul este *generat* de un sistem şi studiul semnalului se reduce la studiul sistemului.

Prezenţa semnalului aleator la intrare necesită folosirea metodelor statistice în studierea acestor sisteme. Atributul *eşantionare* ce caracterizează aceste sisteme are două înţelesuri. În primul rând arată că semnalele de intrare şi ieşire sunt discretizate în domeniul timp şi în domeniul frecvenţă. În al doilea rând am spus că folosim metode statistice pentru studiul acestor sisteme. Dar metodele statistice folosesc un număr relativ mic de date, numit eşantion statistic, pentru a *estima* parametrii semnalelor şi sistemului într-o manieră asemănătoare cu cea folosită la sondajele de opinie. Datorită aplicaţiilor frecvente există o literatură foarte bogată care tratează aceste probleme. În continuare se prezintă doar o introducere în domeniu.

Conducerea automată cu regulatoare numerice este numită *conducere numerică directă (DDC – Direct Digital Control)* şi este caracterizată, în esenţă, de prezenţa calculatoarelor. Datorită principiului de funcţionare al calculatoarelor semnalele folosite la conducerea proceselor sunt prelucrate în mod special. Dintre operaţiile efectuate asupra semnalelor menţionăm eşantionarea, discretizarea, extrapolarea, multiplexarea şi codificarea.

Un exemplu de echipament care foloseşte eşantionarea şi extrapolarea este regulatorul numeric prezentat în Fig. 7.1.

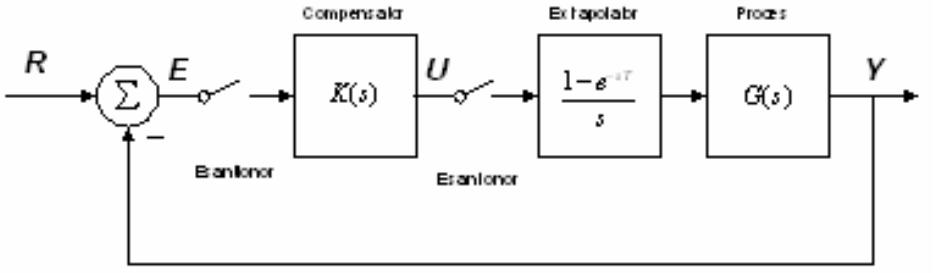


Fig. 7.1 Regulatorul numeric

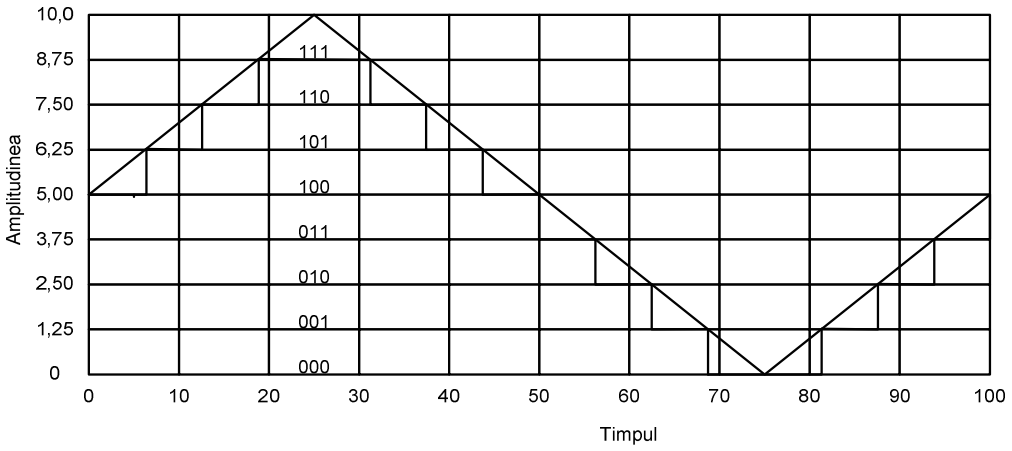


Fig. 7.2 Discretizare în amplitudine cu trei biţi.

7.1. Discretizarea în amplitudine

Pentru a putea fi prelucrate de către calculator semnalele, de exemplu eroarea e(t) din Fig. 7.1, trebuiesc discretizate în amplitudine şi în timp. Dacă semnalul este discretizat de către convertorul analog digital cu ajutorul a trei biţi. există numai 8 nivele discrete de amplitudine şi discretizarea conduce la o aproximare grosieră a semnalului,. Dacă se folosesc 16 biţi în loc de trei se obţin 65536 nivele iar reprezentarea semnalului este mult mai exactă. Numărul de biţi pe care un convertor analog digital îi foloseşte la discretizarea amplitudinii semnalului se numeşte *rezoluţie*.

7.2. Eşantionarea

Discretizarea în timp se numeşte eşantionare. Cu cât frecvenţa de eşantionare este mai mare cu atât reprezentarea semnalului original va fi mai bună. Dacă frecvenţa de eşantionare este mică apar erori. Acestea se manifestă prin faptul că semnalul recuperat după eşantionare apare a avea o frecvenţă complet diferită Fig. 7.3. Această distorsionare a semnalului se numeşte aliere. Frecvenţa de eşantionare minimă este dată de către teorema lui Nyquist care spune că frecvenţa de eşantionare trebuie să fie cel puţin de două ori mai mare decât frecvenţa maximă conţinută în spectrul semnalului. În practică se aleg frecvenţe de eşantionare mult mai mari decât dublul frecvenţei maxime, aşa cum se vede în Fig. 7.4.

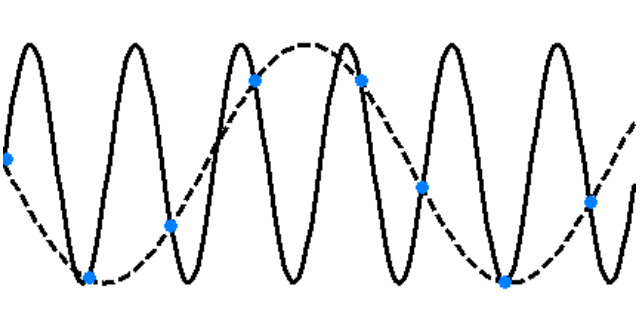


Fig. 7.3 Eşantionarea cu o perioadă prea mare.

O problemă o constituie determinarea frecvenței maxime a spectrului semnalului. Adeseori spectrul semnalului este infinit sau frecvența lui maximă nu este cunoscută.

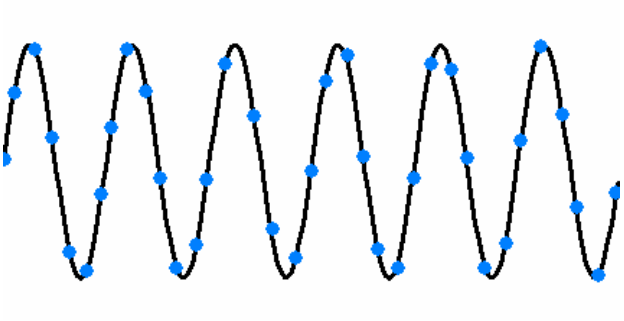


Fig. 7.4 Eşantionarea corectă.

Această dificultate este depăşită printr-o filtrare trece jos hardware a semnalului la intrarea în calculator. Frecvența de tăiere trece jos este luată suficient de mare pentru ca să acopere toate semnalele achiziționate de către convertorul analog digital.

Pentru sistemele cu eşantionare semnalele sunt eşantionate, sau discretizate în timp, cu o perioada constantă T_e, numită perioada de eşantionate.

Pentru a reconstituii un semnal discretizat în timp trebuie satisfăcută teorema de eşantionare a lui Shannon:

$$f_e > 2 f_{\max} \qquad (7.1)$$

în care

$f_e=1/T_e$ este frecvența de eşantionare,

$f_{max}=1/T_{min}$ este frecvenţa maximă din spectrul semnalului.

7.3. Multiplexarea

Multiplexarea reprezintă o metode de a achiziţiona mai multe semnale cu ajutorul unui singur convertor analog digital. Prin multiplexare convertorul eşantionează mai întâi un canal, apoi se comută pe alt canal şi aşa mai departe. Deoarece acelaşi convertor eşantionează succesiv mai multe canale frecvenţa de eşantionare se micşorează proporţional cu numărul canalelor. De exemplu, dacă convertorul are o frecvenţă pentru eşantionarea a 10 canale, de 1.25 Mega eşantioane pe secundă, atunci frecvenţa efectivă de eşantionare a fiecărui canal va fi de 125 kilo eşantioane pe secundă. Prin această tehnică de multiplexare eşantionarea fiecărui semnal se face la momente de timp diferite. Există diferite alte metode de eşantionare şi multiplexare care înlătură acest dezavantaj. Aceasta este necesar atunci când faza semnalului este importantă.

7.4. Transformarea Z a semnalelor eşantionate

În Fig. 7.5 se prezintă eroarea eşantionată din Fig. 7.1. Se observă că aceasta este formată dintr-o serie de impulsuri Dirac $\delta(t)$ distanţate (întârziate) între ele cu perioada de eşantionare T_e. Eroarea eşantionată $e^*(t)$ poate fi exprimată matematic prin relaţia (7.2).

Transformarea Laplace a semnalului de eroare eşantionat cu perioada T_e este dată de (7.3). Se defineşte variabila complexă z în funcţie de variabila complexă Laplace s cu ajutorul relaţiei (7.4).

$$e^*(t)=\sum_{k=0}^{\infty} e(kT_e)\delta(t-kT_e) \tag{7.2}$$

$$E^*(s)=L\{e^*(t)\}=\sum_{k=0}^{\infty} e(kT_e)e^{-ksT_e} \tag{7.3}$$

$$z=e^{sT_e} \tag{7.4}$$

Transformarea Z a semnalului eşantionat $e^*(t)$ este (7.5).

$$E(z)=Z\{e(t)\}=Z\{e^*(t)\}=\sum_{k=0}^{\infty} e(kT_e)z^{-k} \tag{7.5}$$

De exemplu, transformarea Z a semnalului eşantionat

$$x(kT)=0{,}25\delta(t-T)+0{,}5\delta(t-2T) \tag{7.6}$$

este

$$X_e(z) = 0{,}25z^{-1} + 0{,}5z^{-2} \qquad (7.7)$$

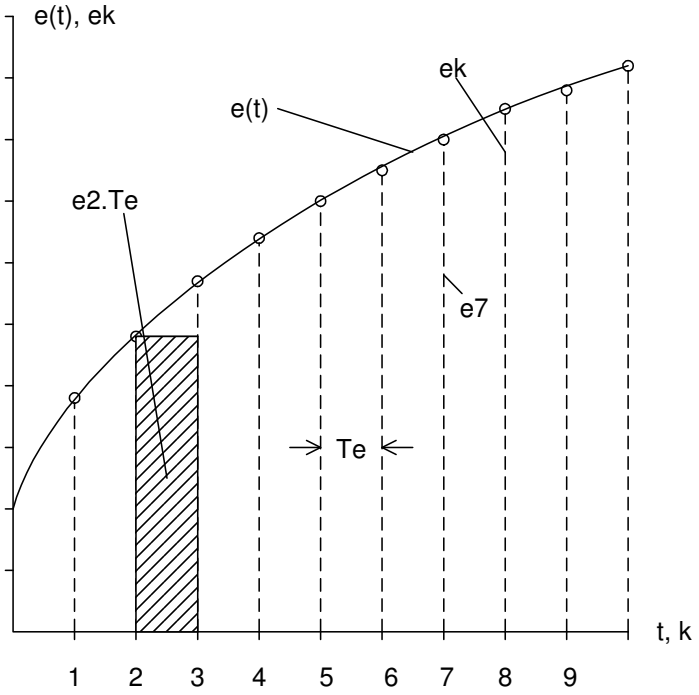


Fig. 7.5 Un semnal eşantionat.

Un alt exemplu, transformarea Z a semnalului treaptă unitară. Deoarece pentru k≥0 avem $u^1(t)=1$ relaţia (7.5) devine:

$$U^1(z) = Z\{u^1(t)\} = \sum_{k=0}^{\infty} z^{-k} \qquad (7.8)$$

Aceasta este o serie geometrică şi poate fi scrisă sub următoarea formă compactă:

$$U^1(z) = \frac{1}{1-z^{-1}} = \frac{z}{z-1} \qquad (7.9)$$

Tab. 7.1

Semnalul $x(t)$	Transformata Laplace $L\{x(t)\}$	Transformata Z $Z\{x(kT_e)\}$
Impuls Dirac $\delta(t)$	1	1
Treapta unitară $u^1(t)$	$\frac{1}{s}$	$\frac{z}{z-1}$
Rampa unitară $u^1(t)t$	$\frac{1}{s^2}$	$\frac{Tz}{(z-1)^2}$
Impuls Dirac întârziat $\delta(t-kT_e)$	$e^{-kT_e s}$	z^{-k}
Semnal întârziat $x(t-kT_e)$	$e^{-kT_e s}X(s)$	$z^{-k}X(z)$

La fel ca şi transformata Laplace, transformata Z se calculează cu ajutorul tabelelor. În Tab. 7.1 se prezintă câteva exemple.

7.5. Funcţia de transfer în domeniul z

Pentru sistemele numerice se poate definii, la fel ca şi pentru sistemele continui, o funcţie de transfer, *Fig. 7.6*. Dacă intrările $x_i(t)$ şi $x_i(kT_e)$ iau aceleaşi valori în momentele de eşantionare, ieşirile $x_e(t)$ şi $x_e(kT_e)$ nu mai iau aceleaşi valori deoarece sistemul reacţionează diferit la un semnal continuu şi la un tren de impulsuri Dirac.

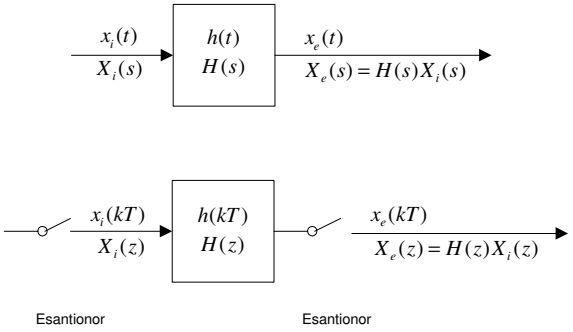


Fig. 7.6 Funcţia de transfer în domeniul s şi în domeniul z.

Să determinăm funcţia de transfer a unui sistem numeric din răspunsul său indicial *Fig. 7.7*. Din figură rezultă expresia analitică a semnalului:

$$y^1(kT) = 0{,}25\delta(t-T) + 0{,}5\delta(t-2T) + 0{,}75\delta(t-3T) + \qquad (7.10)$$
$$+ \delta(t-4T) + \delta(t-5T) + \ldots$$

Transformata Y a răspunsului indicial se calculează cu relaţia (7.5).

$$Y^1(z) = 0{,}25z^{-1} + 0{,}5z^{-2} + 0{,}75z^{-3} + z^{-4} + z^{-5} + z^{-6} + \ldots = \qquad (7.11)$$
$$0{,}25z^{-1} + 0{,}5z^{-2} + 0{,}75z^{-3} + z^{-4}(1 + z^{-1} + z^{-2} + \ldots) =$$
$$0{,}25z^{-1} + 0{,}5z^{-2} + 0{,}75z^{-3} + z^{-4}\frac{1}{1-z^{-1}}$$

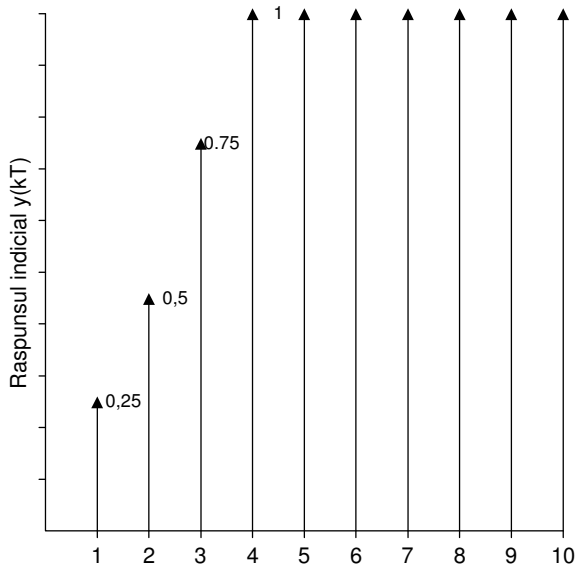


Fig. 7.7 Un răspuns indicial eşantionat determinat experimental.

Deoarece transformata Z a semnalului treaptă unitară este (7.9), din definiţia funcţiei de transfer rezultă:

$$H(z) = \frac{Y^1(z)}{U^1(z)} = (0{,}25z^{-1} + 0{,}5z^{-2} + 0{,}75z^{-3})(1 - z^{-1}) + z^{-4} = \qquad (7.12)$$
$$= 0{,}25(z^{-1} + z^{-2} + z^{-3} + z^{-4})$$

7.6. Funcția de transfer în z echivalentă

Dacă se doreşte ca ieşirea sistemului continuu $x_e(t)$ şi ieşirea sistemului cu eşantionare $x_e(kT)$ să coincidă în momentele de eşantionare se procedează ca în Fig. 7.8 introducând un extrapolator pentru reconstituire semnalului. Efectul extrapolării de ordin zero se vede în *Fig. 7.9*. În intervalul de timp kT şi (k+1)T semnalul refăcut xa(t) are valoarea x(kT).

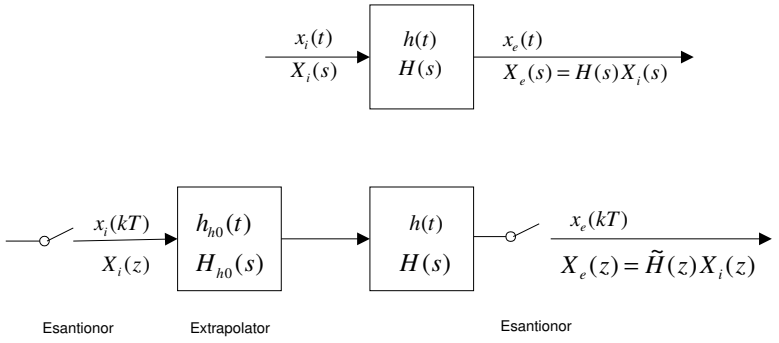


Fig. 7.8 Reconstituirea semnalului prin extrapolare.

Răspunsul extrapolatorului la un singur impuls Dirac, adică funcția pondere, poate fi exprimat în funcție de treapta unitară. dacă considerăm *Fig. 7.9*.

$$h_{h0}(t) = u^1(t) - u^1(t-T) \tag{7.13}$$

Funcția de transfer a extrapolatorului de ordin zero este atunci:

$$H_{h0}(s) = L\{h_{h0}(t)\} = \frac{1}{s} - \frac{e^{-sT}}{s} = \frac{1-e^{-sT}}{s} \tag{7.14}$$

Combinând extrapolatorul de ordin zero cu *H(s)* ca în *Fig. 7.9* rezultă:

$$X_e(z) = \tilde{H}(z)X_i(z) = ZL^{-1}\{H_{h0}(s)H(s)X_i^*(s)\} \tag{7.15}$$

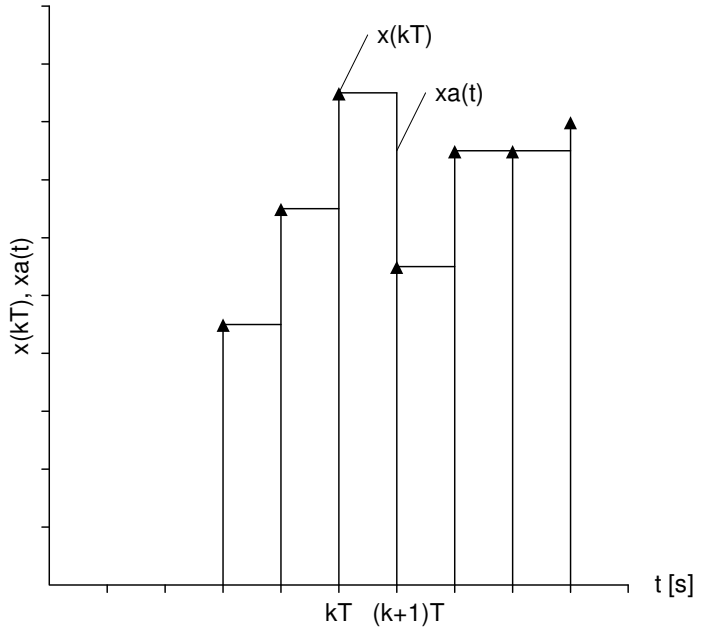


Fig. 7.9 Semnalul aproximat xa(t) cu ajutorul unui extrapolator de ordin zero.

Deoarece $x_i(t)$ este eşantionat şi funcția de transfer a extrapolatorului este dată de (7.14), rezultă:

$$X_e(z) = ZL^{-1}\left\{(1-z^{-1})\frac{H(s)}{s}\right\}X_i(z) \qquad (7.16)$$

Funcția de transfer în z echivalentă este atunci:

$$\tilde{H}(z) = \frac{X_e(z)}{X_i(z)} = (1-z^{-1})ZL^{-1}\left\{\frac{H(s)}{s}\right\} \qquad (7.17)$$

De exemplu, dacă $H(s)=1/s$ este un integrator, rezultă din Tab. 7.1 modelul Euler:

$$H(s)=\frac{1}{s} \quad \rightarrow \quad \tilde{H}(z) = (1-z^{-1})ZL^{-1}\left\{\frac{1}{s^2}\right\} = \frac{T}{z-1} = \frac{Tz^{-1}}{1-z^{-1}} \qquad (7.18)$$

Folosirea unui extrapolator de ordin zero produce, ca un efect lateral, o întârziere cu un timp mort egal cu jumătate din perioada de eşantionare.

$$H(s) = \frac{1-e^{-sT}}{s} \cong \frac{1-1+sT-(sT)^2/2+\ldots}{s} = Te^{-s\frac{T}{2}} \qquad (7.19)$$

7.7. Compensatorul PID numeric

La implementarea numerică a compensatorului PID este foarte frecventă aproximarea acţiunilor integrale şi derivative a algoritmului PID prin metoda diferenţei inverse sau metoda Euler.

$$\textit{Diferenta inversa}: \quad s \cong \frac{1-z^{-1}}{T_e}$$

$$\textit{Euler}: \quad s \cong \frac{1-z^{-1}}{T_e z^{-1}}$$

În aceste relaţii T_e este perioada de eşantionare a semnalului care intră în compensator şi este un parametru care trebuie stabilit de către utilizator alături de constantele K_c, T_i şi T_d.

Aceste aproximări pot fi aplicate la oricare din formele algoritmului PID. Cea mai frecventă variantă constă în folosirea metodei diferenţei inverse la aproximarea acţiunii derivative şi a metodei Euler la aproximarea acţiunii integrale pentru algoritmul PID ideal. Se obţine algoritmul PID numeric de poziţie. Dacă compensatorul comandă, la fiecare perioadă de eşantionare, numai *schimbarea* necesară faţă de poziţia precedentă se obţine algoritmul PID numeric incremental.

$$\textit{PID de pozitie}: \quad K(z) = K_c\left[1+\frac{T_e}{T_i}\frac{z^{-1}}{(1-z^{-1})}+\frac{T_d}{T_e}(1-z^{-1})\right]$$

$$\textit{PID incremental}: \quad K_\Delta(z) = (1-z^{-1})K(z)$$

Compensatorul PID incremental este compatibil cu elemente de execuţie de tip integral, cum ar fi servomotoarele electrice. Acest compensator prezintă avantajul că în caz de avarie elementul de execuţie îşi păstrează ultima poziţie. De asemenea trecerea din regim de comandă manuală în regim de comandă automată se face fără şoc.

Algoritmul PID numeric posedă de cele mai multe ori un mecanism de evitare al efectului de saturare prin acţiunea integrală.

7.8. Instrumentul virtual PID.

Unele regulatoare PID numerice, de exemplu cele realizate pe calculatoare PC sub forma unor instrumente virtuale, au algoritmul PID neliniar. Un exemplu de instrument virtual PID, produs de National Instruments, are structura regulatorului ca în Fig. 6.11. Referinţa r(t) şi variabila de proces $y_t(t)$ la ieşirea traductorului sunt eşantionate cu perioada T_e şi prelucrate de calculator. Eşantionul cu numărul k al semnalului de comandă u(k) al regulatorului este calculat conform variantei de algoritm PID neliniar:

Acţiunea tip P depinde de valoarea constantei L care trebuie să fie aleasă de către utilizator. Comanda u(k) este proporţională cu eroarea e(k) dacă L=1 sau cu pătratul erorii $e^2(k)$ dacă L=0.

Acţiunea tip I realizează integrarea explicită prin metoda trapezului. Şi această acţiune este neliniară. Cu cât eroarea e(i) este mai mare cu atât este mai mică acţiunea integrală.

Acţiunea tip D, realizată de blocul D1, foloseşte metoda diferenţei inverse pentru aproximarea derivatei şi se aplică numai variabilei de proces filtrate $y_f(k)$.

Filtrarea zgomotului N se face de către blocul D2 iar filtrarea referinţei de către blocul F cu ajutorul funcţiilor de transfer din paragraful precedent, în varianta numerică.

$$PID\ neliniar: \quad u(k) = u_P(k) + u_I(k) + u_D(k)$$

$$u_P(k) = K_c e(k)\left[L + (1-L)\frac{|e(k)|}{dom\ r} \right]$$

$$u_I(k) = K_c \frac{T_e}{T_i} \sum_{i=1}^{k} \frac{e(i) + e(i-1)}{2} \cdot \frac{1}{1 + 10\left[\frac{e(i)}{dom\ r} \right]^2}$$

$$u_D(k) = K_c \frac{T_d}{T_e}\left[y_f(k) - y_f(k-1) \right]$$

$$y_f(k) = 0{,}5\, y_t(k) + 0{,}25\, y_t(k-1) + 0{,}175\, y_t(k-2) + 0{,}075\, y_t(k-3)$$

7.9. Modelul ARMAX

Dacă se discretizează intrarea *u(t)* şi ieşirea *y(t)* a unui sistem se vor obţine secvenţa de numere în care *t* este timpul discret normalizat t/T_e.

Relaţia dintre intrarea şi ieşirea unui sistem discret cu eşantionare liniar[160], invariabil[161] în timp este o *ecuaţie cu diferenţe (recurentă).*

[160] Un sistem pentru care este satisfăcut principiul superpoziţiei.
[161] Un sistem pentru care $u(t-T_0)$ produce $y(t-T_0)$.

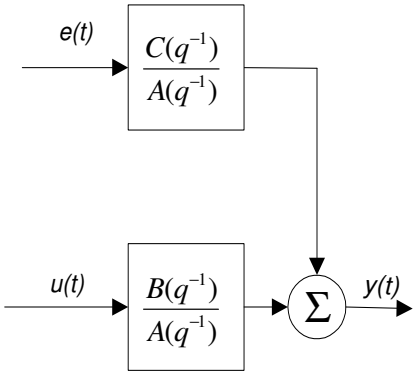


Fig. 7.10 Un model tip ARMAX pentru un sistem discret cu eşantionare.

Un exemplu foarte simplu de sistem cu eşantionare nu are intrarea aleatoare şi este descris de următorul model matematic:

$$y(t) = -a_1 y(t-1) + b_1 u(t-1), \quad \forall t \in N \tag{7.20}$$

în care

u(t) - intrarea în sistem,

y(t) – ieşirea din sistem,

t – timpul discret normalizat cu valori egale cu un număr întreg de perioade de eşantionare T_e.

Modelul matematic (7.20) sub formă de ecuație cu diferențe a sistemului cu eşantionare poate fi obținut pe cale teoretică, din ecuația diferențială a sistemului neeşantionat sau pe cale experimentală, prin măsurarea mărimilor de intrare *u(t)* şi de ieşire *y(t)*.

De exemplu, modelul teoretic al instalației pompă – rezervor din Fig. 3.2 este ecuația diferențială

$$\frac{dy(t)}{dt} = -\frac{1}{T} y(t) + \frac{K}{T} u(t) \tag{7.21}$$

în care u(t) este debitul din conducta care umple rezervorul iar z(t) este nivelul din rezervor.

Prin eşantionare derivata din (7.21) poate fi aproximată prin metoda Euler

$$\frac{dy(t)}{dt} \cong \frac{y(t+T_e) - y(t)}{T_e}$$

Înlocuind în (7.21) se obţin e ecuaţia cu diferenţe

$$y(t+1) = -(\frac{T_e}{T} - 1)\, y(t) + \frac{K}{T} u(t) = -a_1 y(t) + b_1 u(t) \qquad (7.22)$$

Comparând cu (7.20) se obţine identificarea teoretică a modelului sub formă de ecuaţie cu diferenţe a sistemului cu eşantionare:

$$a_1 = \frac{T_e}{T} - 1 \quad , \quad b_1 = \frac{T_e}{T} K \qquad (7.23)$$

Modelul sub formă de ecuaţie cu diferenţe poate fi folosit pentru calculul ieşirii z(t) al sistemului cu eşantionare atunci când se cunoaşte intrarea sa u(t). De exemplu, dacă a_1=*-0,5* iar b_1=*0,5* se obţin rezultatele din următorul tabel pentru răspunsul indicial[162]

t	0	1	2	3	4	5
u(t)	0	1	1	1	1	1
y(t)	0	0,5	0,75	0,875	0,937	0,969

Pentru descrierea mai compactă a modelelor sistemelor cu eşantionare se foloseşte operatorul de întârziere cu o perioadă de eşantionare q^{-1} definit în modul următor:

$$q^{-1} y(t) = y(t-1) \qquad (7.24)$$

Dacă întârzierea este de *d* perioade de eşantionare T_e atunci

$$q^{-d} y(t) = y(t-d) \qquad (7.25)$$

Folosind operatorul de întârziere *q* modelul (7.20) al sistemului cu eşantionare devine

$$(1 + a_1 q^{-1})\, y(t) = b_1 q^{-1} u(t) \qquad (7.26)$$

Un model general, foarte folosit, pentru sistemele cu eşantionare este modelul ARMAX[163] [13], [46] a cărui schemă bloc este prezentată în *Fig. 7.10*. Modelul matematic corespunzător, sub forma unei ecuaţii cu diferenţe, este:

$$y(t) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})} u(t) + \frac{C(q^{-1})}{A(q^{-1})} e(t) \quad , \quad \forall t \in N \qquad (7.27)$$

în care

u(t) – intrarea deterministă a modelului,

[162] Răspunsul sistemului la o intrare treaptă unitară.

[163] Auto-Regresive Moving Average with eXogenous input.

e(t) – intrarea aleatoare a modelului de tip zgomot alb gaussian,
y(t) – ieşirea modelului.
Polinoamele $A(q^{-1})$, $B(q^{-1})$ şi $C(q^{-1})$ au următoarele forme:

$$A(q^{-1}) = 1 + a_1 q^{-1} + ... + a_{na} q^{-na} \tag{7.28}$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + ... + b_{nb} q^{-nb} \tag{7.29}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + ... + c_{nc} q^{-nc} \tag{7.30}$$

Modelele ARMAX sunt folosite la *analiza şi predicţia* seriilor[164] de timp şi la *identificarea experimentală* a funcţiilor de transfer pentru sistemele tehnice[165], organizaţionale, economice sau sociale.

Pentru studiul sistemelor modelul ARMAX caracterizat de (7.27) este uneori simplificat şi prezentat sub denumirile următoare.

Modelul ARX[166]

$$A(q^{-1})\,y(t) = B(q^{-1})u(t) + e(t) \tag{7.31}$$

Modelul FIR[167]

$$y(t) = B(q^{-1})u(t) + e(t) \tag{7.32}$$

Modelul OE[168]

$$y(t) = \frac{B(q^{-1})}{A(q^{-1})}u(t) + e(t) \tag{7.33}$$

Studiul semnalelor sub formă de serii de timp foloseşte şi următoarele modele obţinute din (7.27) prin simplificare.

Modelul ARMA[169]

$$A(q^{-1})\,y(t) = C(q^{-1})e(t) \tag{7.34}$$

Modelul AR[170]

$$A(q^{-1})\,y(t) = e(t) \tag{7.35}$$

[164] Saeria de timp este o colecţie de observaţii efectuate secvenţial în timp.
[165] Instalaţiile.
[166] Autoregresive with exogenous variable
[167] Finite Impulse Response
[168] Output Error
[169] Auto-Regresive Moving Average
[170] Auto-Regressive

Modelele autoregresive de ordinul unu sunt cunoscute şi sub numele de procese Markov. Aceste modele sunt populare deoarece sunt intuitive şi uşor de estimat.

Modelul MA[171]

$$y(t) = C(q^{-1})e(t) \tag{7.36}$$

Pentru studiul sistemelor şi semnalelor sunt folosit şi modele mai complexe decât ARMAX. Printre acestea menționăm:

Modelul ARIMAX[172]

$$y(t) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}u(t) + \frac{C(q^{-1})}{A(q^{-1})\Delta(q^{-1})}e(t) \quad , \quad \forall t \in N \tag{7.37}$$

în care

$$\Delta(q^{-1}) = 1 + q^{-1} \tag{7.38}$$

are un efect integrator asupra perturbației şi permite modelarea comportării ei nestaționare cu polarizare [46].

Modelul NARMAX[173] este un model neliniar spre deosebire de modelul ARMAX care este liniar. Practic aceste modele sunt realizate sub forma unor rețele neurale [7], [46].

7.10. Sisteme cu semnale aleatoare

Perturbațiile sistemelor şi sistemelor automate, vibrațiile, cutremurele şi multe alte semnale pot fi considerate semnale aleatoare. Debitul din sistemul neautomat format dintr-o pompă şi un rezervor poate avea o componentă aleatoare (întâmplătoare, nedeterministă) datorată variațiilor tensiunii de alimentare, sarcinii sau unor surse suplimentare. Valoarea instantanee a unui semnal aleator nu poate fi prevăzută, însă se poate stabili pe cale statistică probabilitatea de realizare a unei anumite valori x, la un moment dat t_1. Se consideră că s-au realizat N înregistrări ale unui semnal aleator x(t). Toate aceste înregistrări formează un *eşantion* statistic de înregistrări cu ajutorul căruia dorim să determinăm parametrii semnalelor aleatoare dintr-o infinitate posibilă de înregistrări care poartă denumirea de *populație.* Printre cel N de înregistrări din eşantion există n înregistrări în care valoarea semnalului x(t) *la momentul* t_1 este mai mică decât valoarea a_1. Dacă numărul total N al valorilor este suficient de mare, atunci se poate calcula probabilitatea $P(a_1,t_1)$ ca semnalul să *nu* depăşească la *momentul* t_1 valoarea a_1:

$$P(a_1, t_1) = \Pr ob[x \le a_1] \cong \frac{n}{N} \tag{7.39}$$

Operația efectuată mai înainte pentru valoarea a_1 a semnalului poate fi repetată pentru orice alte valori x, obținându-se în felul acesta funcția de *repartiție*

[171] Moving Average
[172]. Auto-Regresive Integrated Moving Average with eXogenous input
[173] Nonliniar Auto-Regresive Moving Average with eXogenous input

$P(x, t_1)$. Dacă această funcție admite o derivată, se definește *densitatea de probabilitate* $p(x,t_1)$:

$$p(x,t_1) = \frac{d}{dx} P(x,t_1) \qquad (7.40)$$

care este egală cu probabilitatea ca amplitudinea semnalului să fie cuprinsă între valorile x şi x+dx.

7.10.1. Media şi varianța

Semnalele aleatoare pot fi caracterizate prin diferite *momente de ordin n* definite în modul următor:

$$E[x^n(t_1)] = \int_{-\infty}^{\infty} xp(x)dx \qquad (7.41)$$

Pentru n=1 se obține *media*[174] semnalului μ_x la momentul t_1 :

$$\mu_x = E[x(t_1)] = \int_{-\infty}^{\infty} xp(x)dx \qquad (7.42)$$

Dacă în (7.41) se consideră n=2 se obține *varianța*[175] *(dispersia)* semnalului aleator.

$$\sigma_x^2 = E\left[(x(t_1) - \mu_x)^2\right] = \int_{-\infty}^{\infty} (x-\mu_x)^2 p(x)dx = E[x^2(t_1)] - \mu_x^2 \qquad (7.43)$$

în care σ_x se numeşte *abaterea standard*.

In situațiile practice cele mai frecvente în tehnică mărimile caracteristice semnalelor aleatoare, de exemplu media şi dispersia, sunt constante în timp, adică pentru diferite valori ale lui t_1. Procesele întâmplătoare corespunzătoare acestor semnale se numesc *staționare*. In foarte multe cazuri procesele staționare îndeplinesc şi condiția de *ergocitate* care permite ca operațiile de mediere să se facă în timp pentru o singură înregistrare eşantion cu *numărul k* şi nu pe mai multe eşantioane de semnal la o anumită valoare t_1. In acest caz *media* μ_x se determină cu relația:

$$\mu_x = E[x_k] = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t)dt \qquad (7.44)$$

în care T este durata înregistrării eşantion a semnalului x(t).

Abaterea standard σ_x pentru semnale ergotice se poate determina şi cu relațiile:

[174] Media sau speranța matematică.
[175] Momentul de ordin doi

$$E[x^2{}_k] = \lim_{T\to\infty} \int_{-T/2}^{T/2} x^2(t)dl \qquad (7.45)$$

$$\sigma_x^2 = E\left[(x_k - \mu_x)^2\right] = E[x_k^2] - \mu_x^2 \qquad (7.46)$$

7.10.2. Covarianța şi corelația

Adeseori suntem interesați să aflăm dacă două semnale aleatoare x şi y sunt asociate între ele. În acest scop se calculează covarianța la momentul de timp t_1

$$\sigma_{xy} = E[(x(t_1) - \mu_x)(y(t_1) - \mu_y)] \qquad (7.47)$$

şi corelația

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \qquad (7.48)$$

care este de fapt covarianța normalizată.

Se observă că, prin definiție, corelația dintre două semnale aleatoare are valori cuprinse între +1 şi 1. Semnalele aleatoare independente sunt necorelate şi au coeficientul de corelație zero. Un semnal este perfect corelat cu el însăşi şi deci corelația este unu.

$$\sigma_{xx} = \sigma_x^2 \quad , \quad \rho_{xx} = 1 \qquad (7.49)$$

7.10.3. Eşantionarea statistică şi estimarea parametrilor

În practică nu se pot aplica formulele (7.42) … (7.46) pentru că nu se cunoaşte exact funcția de densitate de probabilitate p(x) şi nici toate valorile necesare ale lui x. De obicei se lucrează cu un *eşantion* statistic de N valori x_i ale lui $x(t_1)$ măsurate la timpul t_1 sau cu N valori discrete x_i în timp[176] ale lui x(t) măsurate pe o singură înregistrare din eşantionul de înregistrări. În acest caz se *estimează* media (7.42) cu o nouă valoare calculată pe un eşantion de înregistrări cu formula:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (7.50)$$

Dacă se extrag mai multe eşantioane[177] cu valori ale lui x_i şi se face media valorilor medii pe eşantion se obține:

---

[176] Timpul normalizat t/T_e, unde T_e este perioada de eşantionare

[177] Eşantioane statistice la timpul t_1 sau eşantioane în timp normalizat pe înregistrarea cu numărul k a eşantionului statistic.

$$E\left[\bar{x}\right] = \mu_x \tag{7.51}$$

În mod asemănător se estimează varianța polarizată la momentul t_1 pentru un eşantion statistic:

$$\text{var}(x) = s_{xp}^2 = \frac{1}{N}\sum_{i=1}^{N}(x(t_1) - \bar{x})^2 \tag{7.52}$$

Media variațelor estimate pe mai multe eşantioane statistice la timpul t_1 se demonstrează că este:

$$E[s_{xp}^2] = \frac{(N-1)}{N}\sigma_x^2 \tag{7.53}$$

Se observă că media valorilor estimate pe eşantioane nu este egală cu valoarea teoretică calculată în întreaga populație cu ajutorul relației (7.43). Se spune că această valoare este *polarizată.* Evitarea polarizării se face calculând varianța cu ajutorul formulei:

$$s_x = \frac{1}{N-1}\sum_{i=1}^{N}(x(t_1) - \bar{x})^2 \tag{7.54}$$

Covarianța s_{xy} şi corelația r_{xy} semnalelor aleatoare x şi y din întreaga populație de înregistrări se estimează pe eşantionul statistic de N înregistrări cu ajutorul formulelor [13]:

$$\text{cov}(x, y) = s_{xy} = \frac{\sum_{i=1}^{N}(x(t_1) - \bar{x})(y(t_1) - \bar{y})}{N} \tag{7.55}$$

$$r_{xy} = \frac{s_{xy}}{s_x s_y} \tag{7.56}$$

7.10.4. Un exemplu de semnal aleator

Un exemplu simplu de semnal aleator x(t) este semnalul aleator cu valori distribuit uniform în intervalul [a,b].Pentru acesta la momentul t_1 variabila aleatoare $x(t_1)$ ia valori în intervalul [a,b] care apar cu aceiaşi probabilitate. In acest caz densitatea de probabilitate este:

$$p(x) = \frac{1}{b-a} \tag{7.57}$$

Media semnalului la timpul t_1 va fi:

$$\mu = \frac{b-a}{2} \tag{7.58}$$

Abaterea standard a semnalului faţă de medie la timpul t_1 va fi:

$$\sigma = \frac{b-a}{\sqrt{12}} = 0{,}288675.(b-a) \tag{7.59}$$

7.10.5. Intervale de încredere

Dacă se cunoaşte densitatea de probabilitate a semnalului se pot calcula intervalele de încredere pentru media şi dispersia semnalului calculate pe eşantionul statistic [8].

Semnalele aleatoare întâlnite în practică admit cel mai frecvent o funcţie de repartiţie denumită ***normală*** sau a lui ***Gauss***. Funcţia densităţii de probabilitate corespunzătoare este definită prin expresia:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{(x-\mu)^2}{2\sigma^2}] \tag{7.60}$$

Importanţa deosebită a repartiţiei normale este evidenţiată de ***teorema limitei centrale*** care afirmă că suma unui număr mare de mărimi aleatoare independente are, în condiţii suficient de generale, o densitate de probabilitate apropiată de cea normală.

Folosind definiţia densităţii de probabilitate normale rezultă funcţia de repartiţie normală:

$$P(x) = \frac{1}{\sigma.\sqrt{2\pi}} \int_{-\infty}^{x} \exp[-\frac{(z-\mu)^2}{2\sigma^2}].dz \tag{7.61}$$

Cu ajutorul acestei relaţii se determină probabilitatea ca un semnal aleatoriu normal să aibă valori cuprinse într-un anumit interval. De exemplu, probabilitatea ca semnalul x(t) să ia valori cuprinse în intervalul [-cσ, cσ] în jurul valorii medii μ are valoarea:

c=1 P=68,3%

c=2 P=95,4%

c=3 P=99,7%

Se observă că practic semnalul aleator normal (gaussian) se modifică în jurul mediei μ cu o valoare mai mică de trei abateri standard σ.

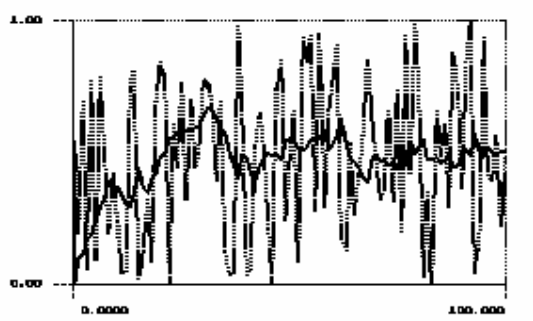


Fig. 7.11. a) Variația în timp a unui semnal aleatoriu cu densitatea de probabilitate p(x) constantă şi egală cu 1 în intervalul (0,1). b) Efectul filtrării semnalului cu un filtru trece jos.

7.10.6. Filtrarea

În subdirectorul *Demo* al KitSAS[178] se găseşte programul *Aleator.sem* care permite simularea unui semnal aleator uniform distribuit în intervalul [0,1]. În Fig. 7.11 se prezintă simularea acestui semnal.

Programul permite determinarea mediei şi abaterii standard a semnalului. care au următoarele valori:

$$\mu=0,5$$
$$\sigma=0,288675$$

Se pot compara valorile obținute prin simulare cu ajutorul kitului cu cele teoretice obținute anterior.

Rezultatul trecerii semnalului aleator uniform distribuit printr-un filtru trece-jos cu funcția de transfer:

$$H(s)=\frac{1}{5.s+1} \qquad (7.62)$$

se observă în Fig. 7.11 Caracteristicile măsurate de kit sunt de data aceasta următoarele:

$$\text{media } \mu = 0,5$$
$$\text{abaterea standard } \sigma = 0,076$$

Şi în Fig. 7.11 se observă că media rămâne aceiaşi dar împrăştierea în jurul mediei se micşorează mult.

[178] Kitul de Simulare şi Analiză a Sistemelor este prezentat în paragraful 10.1

7.10.7. Analiza semnalelor aleatoare.

Semnalele aleatoare pot avea *forme* foarte diferite după cum se vede în Fig. 7.12 In scopul punerii în evidenţă a formei unui semnal aleatoriu trebuie stabilită o relaţie între valorile lui la diferite momente de timp. Pentru cazul a două momente de timp această relaţie este funcţia de repartiţie de ordinul doi $P(x_1, x_2 ; t_1, t_2)$ care se defineşte în mod asemănător cu cea de ordinul întâi, drept probabilitatea ca semnalul aleatoriu să nu depăşească valoarea x_1 la timpul t_1 şi valoarea x_2 la timpul t_2. Pentru cazul a N înregistrări această probabilitate poate fi determinată aproximativ raportând numărul cazurilor în care semnalele sunt mai mici decât a_1 şi a2 , pentru timpii t_1 şi t_2 , la numărul total de cazuri N. Funcţiei de repartiţie de ordinul doi îi corespunde o funcţie a densităţii de probabilitate de ordinul doi definită astfel:

$$p(x_1, x_2; t_1, t_2) = \frac{\partial P(x_1, x_2; t_1, t_{2)}}{\partial x_1 \partial x_2} \qquad (7.63)$$

7.10.8. Funcţia de autocorelaţie.

Un semnal aleator va fi definit complet când vor fi cunoscute toate densităţile de probabilitate de ordinul 1,2,3,…,n. Apare evident că determinarea experimentală şi manipularea matematică a funcţiilor de densitate de probabilitate este dificilă. Din fericire, pentru cazurile practice cele mai frecvente ale semnalelor staţionare şi gaussiene este suficientă cunoaşterea densităţilor de probabilitate de ordinul unu şi doi. Pentru determinarea densităţii de probabilitate gaussiene de ordinul doi este necesară şi cunoaşterea unei noi caracteristici denumită funcţia de autocorelaţie:

$$R(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t).x(+\tau).dt \qquad (7.64)$$

în care T este durata înregistrării.
Pentru $\tau = 0$ se obţine:

$$R(0) = E[x^2] \qquad (7.65)$$

iar pentru $\tau = \infty$ se obţine:

$$R(\infty) = \mu^2 \qquad (7.66)$$

Funcţia de autocorelaţie a unui semnal determinist periodic este o funcţie periodică cu aceiaşi perioadă. In cazul semnalelor gaussiene cunoaşterea funcţiei de autocorelaţie le determină complet proprietăţile statistice deoarece μ şi σ pot fi determinate din *R(0)* şi *R(∞).* Modelele matematice ale semnalelor aleatorii

prezentate până acum au fost elaborate în domeniul timp. Modele echivalente, care prezintă unele avantaje în privința obținerii datelor experimentale şi prelucrării matematice, pot fi şi în domeniul frecvență.

7.10.9. Densitatea spectrală de putere.

Se defineşte *densitatea spectrală a puterii*:

$$G(f) = 2.\lim_{T\to\infty}\frac{1}{T}E[|X(f,T)|^2] \qquad (7.67)$$

în care X(f, T) este transformarea Fourier finită a funcției x(t):

$$X(f,T) = \int_0^T x(t).\exp[-j2\pi ft].dt \qquad (7.68)$$

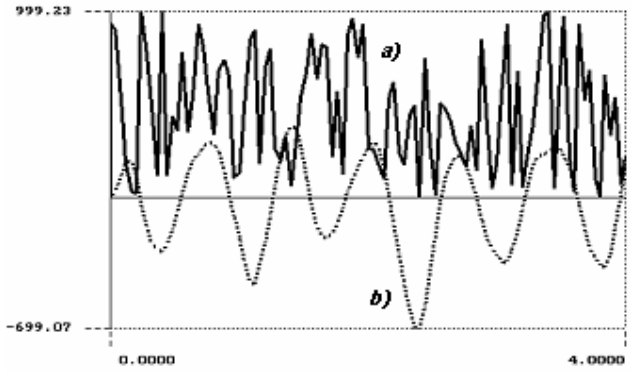


Fig. 7.12. Un semnal aleator cu banda largă de frecvență a) şi rezultatul filtrării sale de către un oscilator mecanic b).

O estimare a densității spectrale de putere G(f) poate fi obținută în mod simplu ştergând în relația de definiție semnul limită şi semnul speranței (medierii statistice) matematice E:

$$\tilde{G}(f) = \frac{2}{T}|X(f,T)|^2 \qquad (7.69)$$

Sistemul KitSAS determină direct modulul transformării Fourier. Pentru obținerea estimației de mai sus modulul trebuie ridicat la pătrat şi înmulțit cu 2/T. În subdirectorul *Demo* se găseşte programul *Rand_or2.s* care simulează un semnal aleatoriu cu banda largă de frecvență, cum ar fi de exemplu un cutremur, determină răspunsul unui oscilator mecanic la acest semnal *Fig. 7.12* şi îi calculează modulul transformării Fourier în *Fig. 7.13* Oscilatorul are pulsația naturală $\omega_n = 10$, adică f_n

= 1,59 Hz. Se observă că densitatea spectrală de putere are maximum la această frecvenţă. De asemenea curba are un caracter aleator. Pentru obţinerea unei valori mai exacte sunt necesare mai multe estimări care apoi se mediază.

Se poate demonstra că puterea medie P a semnalului aleator are valoarea:

$$P = R(0) = \lim_{T\to\infty} E[x^2(t)] = \int_0^\infty G(f).df \qquad (7.70)$$

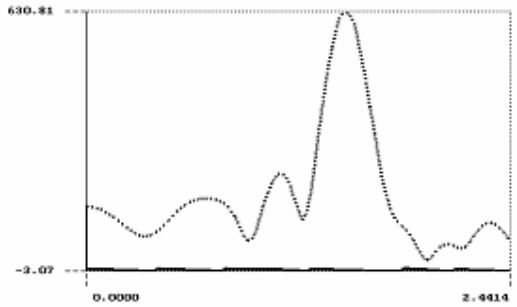


Fig. 7.13. Modulul transformării Fourier a răspunsului oscilatorului mecanic la vibraţii aleatoare cu banda largă de frecvenţă.

Pentru vibraţii aleatorii cu banda îngustă de frecvenţă, aşa cum este cazul în Fig. 7.13 densitatea spectrală de putere este constantă şi egală cu G_0 în intervalul Δf. Integrala care determină puterea se calculează simplu:

$$P_1 = G_1.\Delta f \qquad (7.71)$$

de unde rezultă o nouă definiţie a densităţii spectrale de putere:

$$G = \frac{P}{\Delta f} \qquad (7.72)$$

Această relaţie justifică denumirea de densitate spectrală de putere dată lui G(f).

Densitatea spectrală de putere este transformarea Fourier a funcţiei de autocorelaţie.

$$G(f) = 2\int_{-\infty}^{\infty} R(\tau)\exp[-2\pi jf\tau]d\tau \qquad (7.73)$$

pentru $0<f<\infty$.

7.10.10. Răspunsul sistemelor la semnale aleatorii.

Dacă intrarea u(t) într-un sistem liniar este aleatorie atunci ieşirea y(t) este tot aleatorie şi depinde de intrare prin intermediul relației:

$$G_y(f) = |H(f)|^2 \, G_u(f) \qquad (7.74)$$

în care

$G_u(f)$ - densitatea spectrală de putere a semnalului de intrare,

$G_y(f)$ - densitatea spectrală de putere a semnalului de ieşire,

H(f) - funcția de transfer a sistemului

De exemplu, să considerăm că debitul de intrare în rezervorul tampon este un semnal aleatoriu de tip zgomot alb cu densitatea spectrală de putere constantă şi egală cu *a* şi cu media $\mu_u = 0$. Folosind relația precedentă se calculează densitatea spectrală de putere a semnalului de ieşire care în acest caz este nivelul în rezervor.

$$G_y(\omega) = \left| \frac{K_F}{1 + T_F j\omega} \right|^2 \cdot a = \frac{K_F^2}{1 + T_F^2 \omega^2} \cdot a \qquad (7.75)$$

Valoarea medie a ieşirii, la fel ca şi a intrării, este nulă:

$$\mu_y = 0 \qquad (7.76)$$

Dispersia semnalului în jurul valorii medii este:

$$\sigma^2 = E[y^2 - \mu_y^2] = E[y^2[= \int_{-\infty}^{\infty} y^2 p(y) dy$$

$$\sigma^2 \cong \frac{1}{T} \int_0^T y^2(t) dt = \int_0^{\infty} G_y(f) df = \frac{1}{2\pi} \int_0^{\infty} G_y(\omega) d\omega$$

$$\sigma^2 = \frac{aK_F}{2\pi} \int_0^{\infty} \frac{d\omega}{1 + T_F \omega^2} = \frac{aK_F}{2\pi T_F} \int_0^{\infty} \frac{dx}{1 + x^2} = \frac{aK_F}{4T_F} \qquad (7.77)$$

Se observă că dacă constanta de timp T_F se dublează, dispersia devine $0{,}7\sigma$ adică se micşorează cu numai 30%. Un sistem de reglare automată a nivelului poate fi o soluție mai bună.

7.11. Eşantionarea în domeniul frecvenţă.

Orice semnal obţinut prin achiziţie de date sau simulare poate fi analizat în domeniul frecventa. In acest scop analizatorul de frecventa din al kitului determina transformarea Fourier discretă a semnalului cu ajutorul algoritmului FFT (Fast Fourier Transform). Transformarea Fourier discretă stabileşte legătura dintre semnalul în domeniul timp x(t) discretizat şi semnalul în domeniul frecventa discretizat X(f):

$$x(n.dt) \leftrightarrow X(k.df)$$

în care

dt - pasul de discretizare în domeniul timp,

df - pasul de discretizare în domeniul frecvenţă.

Relaţia dintre transformarea Fourier discretă X(k.df), *determinată de kit,*, transformarea Fourier rapidă si coeficientul seriei Fourier este:

$$X(k \,.\, df) = dt \,.\, Xk = N \,.\, dt \,.\, Ck$$

în care

X(k .df) - componenta k a transformării Fourier discrete,

Xk - componenta k a transformării Fourier rapide,

Ck - coeficientul k al seriei Furier,

dt - pasul de discretizare al semnalului x(n.t),

N - numărul de paşi de discretizare dt într-o perioadă de calcul a transformării Fourier rapide.

In cazul transformării Fourier discrete atât semnalul în domeniul timp x(t) cât si modulul semnalului în domeniul frecventa |X(f)| trebuie să fie funcţii periodice. Pentru aplicarea algoritmului transformării Fourier rapide numărul N de paşi de discretizare dt din perioada T0 a semnalului trebuie să fie o putere a lui 2:

$$N = 2\text{^}R$$

în care R = 0, 1, 2, 3, ...

In versiunea 3.11 a programului KitSAS valoarea maximă a lui R este 13. Deci valorile posibile pentru N sunt următoarele:

R	**N**
0	1
1	2
2	4
3	88
4	16
5	32
6	64
7	128

8	256
9	512
10	1024
11	2048
12	4096
13	8192

Perioada semnalului poate să aibă numai valorile date de relația:

$$T0 = N \cdot dt$$

Deoarece semnalul x(t) este periodic rezultă că spectrul de amplitudini |X(f)| este discret cu pasul df determinat de relația:

$$df = 1 / T0$$

De exemplu, dacă pasul de eşantionare a semnalului x(t) este dt = 1 sec si se alege valoarea R=10 atunci N=1024 şi

$$df = 1 / 1024\ Hz \sim 1\ mHz$$

Reciproc, dacă spectrul de amplitudini |X(f)| este periodic, cu perioada S0 , atunci semnalul x(t) este discret, cu pasul:

$$dt = 1 / S0$$

Modulul şi faza funcţiei de transfer la perturbaţie a sistemului automat de stabilizare cu regulator proporţional. Se determină frecvenţa de bandă f_B la −3 dB

Numarul de date ales N = 4096

	Modulul	Faza
0	3.33300870369974E-0001	0.00000000000000E+0000
1	3.15610420119860E-0001	−3.28610993417171E-0001
2	2.75585428208039E-0001	−5.98871857407995E-0001
3	**2.33339104459792E-0001**	**−7.97773603006360E-0001**
4	1.97400449092470E-0001	−9.40008145567880E-0001
5	1.68980360435136E-0001	−1.04306050894727E+0000
6	1.46664589778166E-0001	−1.11976400015118E+0000

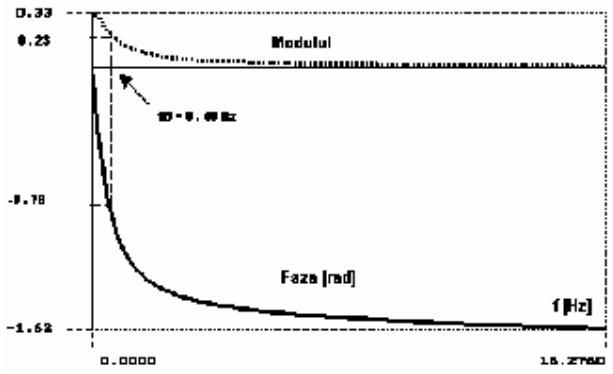


Fig. 7.14. Diagramele Bode ale funcţiei de transfer la perturbaţie a sistemului automat de stabilizare cu regulator proporţional.

Pentru a putea fi prelucrate numeric x(t) şi |X(f)| sunt discretizate cu paşii dt în domeniul timp şi df în domeniul frecvență. Această discretizare implică periodizarea funcţiilor x(t) si |X(f)| cu perioadele T0 si S0 în cele două domenii, de timp si de frecventa. Dar pentru a putea periodiza |X(f)| aceasta trebuie să aibă o lungime finită, adică o frecventa maximă Fmax. Deoarece |X(f)| este o funcţie pară, adică,

$$|X(f)| = |X(-f)|$$

rezultă că se poate face o periodizare cu durata cea mai mică

$$S0 >= 2 . Fmax$$

De aici decurge teorema eşantionării a lui Shannon care determină pasul de eşantionare cel mai mare :

$$dt = 1 / S0 <= 1 / (2 . Fmax)$$

In practică se alege deseori:

$$t = 1 / (10 .. 20) . Fmax$$

Spectrul de amplitudini |X(f)| are si el în practică cel puţin 10 .. 20 de puncte. Deci

$$Fmax >= (10 ..20) . df >= (10 .. 20) / T0$$

Înlocuind în relaţia precedentă rezultă valoarea aproximativă:

$$dt <= T0 / 400$$

Deoarece perioada T0 poate fi împărţită numai într-un număr de paşi N=2^R rezultă că

$$R >= 9$$

In concluzie, o regulă bună de pornire la analiza în domeniul frecvență a unui semnal indică discretizarea sa în cel puţin 16 sau 32 de paşi si alegerea unei periodizări cu T0 egal cu cel puţin 512 paşi, adică R= 9.

7.12. Regulile de eşantionare ale KitSAS

Pentru stabilirea perioadelor de eşantionare în domeniul timpului şi în domeniul frecvenţei este necesară cunoaşterea duratei, măcar aproximative, a procesului tranzitoriu a procesului simulat.

- **Prima regulă:** Pentru o simulare rapidă se alege un pas de calcul de o sută de ori mai mic decât durata procesului tranzitoriu:

  $$dt = Pas = (durata\ procesului\ tranzitoriu) / 100$$

- **A doua regulă:** Pentru o analiză rapidă, şi deci mai puţin precisă, în domeniul frecvenţă se periodizează ieşirea cu perioada:

  $$T_0 = N.dt = 512.dt$$

  deci R = 10

  Rezultă perioada de eşantionare în domeniul frecvenţă:

  $$df = 1 / T_0 = 1 / (512.dt)$$

- **A treia regulă:** O simulare de precizie necesită un pas de calcul:

  $$dt = Pas = (durata\ procesului\ tranzitoriu) / 2000$$

- **A patra regulă:** periodizarea pentru o analiză de precizie în domeniul frecvenţă se face cu

  o valoare R = 12 , adică:

  $$T_0 = 4096.dt$$

  şi perioada de eşantionare în domeniul frecvenţă este:

  $$df.= 1 / (4096.dt)$$

Aplicând aceste reguli pentru o analiză cu erori cât mai mici în domeniul frecvenţă se obţine pentru o durată a procesului tranzitoriu de 3 secunde:

dt = Pas de calcul = 3/2000 = 0.0015 s

R = 12

df = 1/(4096.0,0015) = 0,16276

7.13. Identificarea sistemelor

Identificarea sistemelor reprezintă abordarea experimentală a modelării proceselor. Prin identificare se pot obţine modele neparametrice şi modele parametrice. Exemple de modele neparametrice sunt răspunsul indicial şi diagramele Bode. Modele parametrice obţinute prin identificare sunt funcţia de transfer, ecuaţia diferenţială şi ecuaţia cu diferenţe. În continuare ne vom referi numai la identificare parametrică folosind modele de tip ARMAX.

Identificarea se face în cinci etape:

- Proiectarea unui experiment prin care se achiziţionează semnalele de intrare şi ieşire din sistem. În această etapă se stabileşte care sunt variabilele care se măsoară, tipul lor (continuu, logic, discret) şi protocolul de experimentare.
- Se alege structura modelului. În acest scop se folosesc cunoştinţe obţinute înaintea experimentului şi se fac alegeri prin încercări.

- Un criteriu de apreciere al gradului de potrivire al modelului la datele experimentale este stabilit în această etapă. Se caută optimizarea identificării după acest criteriu.
- Se estimează parametrii modelului.
- Se validează structura şi valorile parametrilor modelului.

Modelele parametrice ARMAX folosite în continuare pentru identificare prezintă numeroase avantaje.

1. Se aplică pentru diferite tipuri de sisteme: tehnice, organizaționale, economice, sociale, financiare, psihologice, etc
2. Semnalele folosite pentru identificare pot fi de amplitudine mică.
3. Modelul obținut are o precizie ridicată.
4. Influența perturbațiilor este redusă.
5. Timpul de calcul este redus.
6. Se poate urmării variația parametrilor în timp real.

Identificarea sistemelor prin modele ARMAX se poate reduce la o problemă de *regresie liniară*. Se poate arăta aceasta folosind un exemplu foarte simplu de model ARMAX (7.27) în care A'1, B=b_0 şi C=1. Relația intrări – ieşire pentru model se reduce în acest caz la

$$y = b_0 u + e \qquad (7.78)$$

Pentru ca sistemul să fie liniar este necesar ca dreapta (7.78) care reprezintă caracteristica sa statică să treacă prin origine, fapt care poate fi realizat printr-o alegere corespunzătoare a sistemului de referință. În cazul general însă ecuația dreptei va avea forma

$$(y - y_0) = b_0 (u - u_0) + e \qquad (7.79$$

sau

$$y = a + bu + e \qquad (7.80)$$

Se pot estima parametrii *a* şi *b* ai acestui model cu ajutorul unui eşantion de *N* date *[x_i , y_i]* obținându-se valorile *a'* si *b'*. Pentru ca relația (7.80) să rămână valabilă pentru valorile din eşantion şi parametrii estimați trebuie să se înlocuiască semnalul aleator de eroare *e* cu reziduul r_i.

$$y_i = a' + b' u_i + r_i \qquad (7.81$$

Parametrii acestui model pot fi estimați prin metoda celor mai mici pătrate care impune ca suma pătratelor reziduurilor să fie minimă.

$$J = \sum_{i=1}^{N} r_i^2 = \sum_{i=1}^{N} (y_i - a' - b' u_i)^2 = \min \qquad (7.82)$$

Pentru aceasta este necesar ca

$$\frac{\partial J}{\partial a'} = 0 \qquad (7.83)$$

şi

$$\frac{\partial J}{\partial b'} = 0 \tag{7.84}$$

Calculând condiția (7.83) rezultă valoarea *a'* care este estimația parametrului *a*.

$$\frac{\partial J}{\partial a'} = \sum_{i=1}^{N} 2(y_i - a' - b'u_i) = 0 \tag{7.85}$$

$$-Na' + \sum_{i=1}^{N} (y_i b'u_i) = 0 \tag{7.86}$$

$$a' = \frac{1}{N}\sum_{i=1}^{N} y_i - b'\frac{1}{N}\sum_{i=1}^{N} u_i = \overline{y} - b'\overline{u} \tag{7.87}$$

Calculând în mod asemănător condiția (7.84) rezultă estimația b' a modelului:

$$b' = \frac{\sum_{i=1}^{N} (u_i - \overline{u}) y_i}{\sum_{i=1}^{N} (u_i - \overline{u})^2} \tag{7.88}$$

8. Sisteme inteligente cu logica fuzzy

Funcţionarea sistemelor în medii ostile, aşa cum este cazul echipamentelor şi instalaţiilor pentru clădiri şi locuinţe, conduce, printre altele, la existenţa unui nivel ridicat de perturbaţii şi zgomote. Dacă zgomotele depăşesc aproximativ 10 – 20 % din valoarea semnalului util identificarea funcţiei de transfer a procesului nu mai este exactă şi conducerea adaptivă nu mai dă rezultate mulţumitoare. În această situaţie se folosesc pentru conducere sisteme cu inteligenţă artificială cum ar fi sistemele expert, sistemele cu logică fuzzy, reţelele neurale şi sistemele cu algoritmi genetici.

Există multe discuţii asupra definiţii sistemelor inteligente. Într-o accepţiune extinsă se acceptă că sistemele expert, sistemele cu logică fuzzy şi sistemele cu algoritmi genetici fac parte din această categorie. În Tab 8.1 se face o comparaţie a principalelor caracteristici a acestor sisteme. O examinare sumară a tabelului pune în evidenţă avantajele reţelelor neurale şi sistemelor cu algoritmi genetici. Trei performanţe critice sunt reprezentarea cunoştinţelor, învăţarea şi explicarea acţiunilor.

*Tab 8.1 Compararea Sistemelor Expert (**SE**), Sistemelor Fuzyy (**SF**), Reţelelor Neurale (**RN**) şi Algoritmilor Genetici (**AG**).*

	SE	***SF***	***RN***	***AG***
Reprezentarea cunoştinţelor	**	***		**
Toleranţă la incertitudine	**	***	***	***
Toleranţă la imprecizie		***	***	***
Adaptabilitate		**	***	***
Învăţare			***	***
Explicarea acţiunilor	***	***		*
Descoperirea cunoştinţelor		*	***	**
Întreţinere		**	***	**

Teoria mulţimilor fuzzy (fuzzy) a fost elaborată în Statele Unite ale Americii la începutul anilor şaizeci de către matematicianul Lotfi Zadeh, profesor la Berkeley University din California. Au fost necesare aproximativ treizeci de ani pentru ca această teorie să fie recunoscută în domeniul automaticii, mai ales datorită faptului că în această perioadă, logica clasică a stat la baza unei avalanşe de aplicaţii în domeniul conducerii automate. Mulţimile fuzzy şi aplicaţiile lor au constituit preocupări importante şi pentru cercetătorii români, dintre care se remarcă în special C.V. Negoiţă.

Logica fuzzy, este o logică polivalentă, spre deosebire de logica lui Boole care este bivalentă (binară). Diferenţa între logica clasica, unde nu se raţionează decât la modul "tot sau nimic", şi logica "fuzzy", este că aceasta din urmă ia în considerare o infinitate de situaţii intermediare de tipul: tot, mult, mijlociu, puţin, foarte puţin, nimic.

Raţionamentele oamenilor sunt foarte rar binare, deoarece noţiunile de "adevărat" sau "fals" în mod strict, apar numai uneori. Mult mai frecvent se folosesc noţiunile "poate" şi "mai degrabă adevărat sau fals".

Aplicaţiile bazate pe logică fuzzy nu au fost posibile până relativ recent, când tehnica de calcul a devenit performantă şi faptul că logica fuzzy necesită mai multe calcule decât logica binară, nu mai reprezintă un obstacol în abordarea aplicaţiilor.

Conducerea automată fuzzy a proceselor continue au început să fie cercetate din 1974 de către Mamdani, profesor la Queen Mary College din Marea Britanie. Munca sa de pionierat a fost motivată de două publicaţii ale lui Zadeh despre algoritmii fuzzy (în anul 1968) şi despre analiza lingvistică (în anul 1973).

Un alt cercetător important care a adus contribuţii remarcabile în conducerea fuzzy este Michio Sugeno, de la Departament of System Science, Tokyo Institute of Technology din Japonia.

Termenul "fuzzy" din limba engleză, care înseamnă fuzzy sau imprecis, este folosit în domeniul de specialitate şi în limba română.

8.1.1. Mulţimi fuzzy.

O mulţime fuzzy poate fi considerată ca fiind o categorie de obiecte unde nu există delimitări foarte clare între obiectele care aparţin şi cele care nu aparţin categoriei. Pentru mulţimea fuzzy se poate folosi definiţia prezentată în cele ce urmează.

Fie X ={x} o grupare de elemente desemnate generic prin *x* şi numită *mulţime de bază* sau *univers*.

Atunci mulţimea fuzzy A (submulţime a universului X) este o mulţime de perechi ordonate

$$A = \{(x, \mu_A(x))\}, \quad x \in X \tag{8.1}$$

unde termenul $\mu_A(x)$ este numit *gradul de apartenenţă* a lui *x* la mulţimea A şi μ_A: X→M este o funcţie pe X cu valori în M, denumit *spaţiu de apartenenţă*. Atunci când M conţine numai două valori 0 şi 1, mulţimea A este nefuzzy şi funcţia de apartenenţă devine identică cu funcţia caracteristică unei mulţimi nefuzzy.

De obicei, se presupune că M este intervalul [0,1], cu 0 şi 1 reprezentând cel mai mic, respectiv cel mai mare grad de apartenenţă. O mulţime fuzzy A poate fi definită cu precizie prin asocierea fiecărui element *x* a unui număr cuprins între 0 şi 1 care reprezintă gradul său de apartenenţă la mulţimea A.

În cadrul teoriei clasice a mulţimilor, funcţia de apartenenţă $\mu_A(x)$ a unui element *x* la o mulţime A care este o submulţime a universului X, este definită astfel:

$$\mu_A(x) = \begin{cases} 1 \text{ dacă } x \in A \\ \\ 0 \text{ dacă } x \in A \end{cases} \qquad (8.2)$$

Se poate spune că elementul *x* este membru al mulţimii A dacă are funcţia de apartenenţă $\mu_A(x)=1$ sau nu este membru al mulţimii A dacă are funcţia de apartenenţă $\mu_A(x)=0$.

Pentru o mulţime fuzzy, apartenenţa este exprimată în intervalul [0,1] şi nu este exprimată în mulţimea {0,1}. Mulţimea fuzzy A, definită ca o submulţime fuzzy a mulţimii X, este caracterizată de valorile funcţiilor de apartenenţă pentru toate elementele x_i ale mulţimii X:

$$A = \sum_{i=1}^{m} \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \cdots + \mu_A(x_m)/x_m \qquad (8.3)$$

Membrul drept al relaţiei se poate citi astfel: *elementul x_1 cu funcţia de apartenenţă la mulţimea fuzzy A având valoarea $\mu_A(x_1)$ şi elementul x_m cu funcţia de apartenenţă la mulţimea fuzzy A având valoarea $\mu_A(x_m)$.*

Dacă universul X este continuu, atunci mulţimea fuzzy se notează

$$A = \int_X \mu_A(x)/x \qquad (8.4)$$

ceea ce desemnează mulţimile fuzzy în general.

- ***Operatii cu mulţimi fuzzy.***

Pentru mulţimile fuzzy sunt definite operaţii similare celor definite pentru mulţimile clasice. Dar, spre deosebire de mulţimile clasice, operatorii nu sunt unic definiţi, datorită valorii continue a funcţiei de apartenenţă în intervalul [0,1].

Extinderea operaţiilor de reuniune şi intersecţie în domeniul mulţimilor fuzzy nu este unic definită. Deoarece mulţimile clasice pot fi considerate mulţimi fuzzy speciale, operaţiile de reuniune şi de intersecţie trebuie să fie aceleaşi dacă mulţimile fuzzy devin mulţimi clasice.

Intersecţia. Intersecţia mulţimilor fuzzy A şi B este desemnată prin $A \cap B$ şi este definită ca o mulţime fuzzy conţinută deopotrivă în A şi B. Funcţia de apartenenţă pentru $A \cap B$ este dată de

$$\mu_{A \cap B}(x) = min\,(\mu_A(x), \mu_B(x)), \qquad x \in X \qquad (8.5)$$

unde Min(a, b) = a dacă $a \le b$ şi Min(a, b) = b dacă $a > b$. Dacă se foloseşte simbolul $\wedge$ pentru conjuncţie în locul Min, relaţia (9.5) se poate scrie

$$\mu_{A \cap B} = \mu_A \wedge \mu_B \qquad (8.6)$$

Reuniunea. Noţiunile de reuniune şi intersecţie a mulţimilor A şi B sunt duale. Reuniunea mulţimilor A şi B este desemnată prin A∪B şi este definită ca fiind cea mai mică mulţime fuzzy care conţine deopotrivă mulţimile A şi B. Funcţia de apartenenţă pentru A∪B este dată de

$$\mu_{A \cup B}(x) = max\,(\mu_A(x), \mu_B(x)), \qquad x \in X \qquad (8.7)$$

unde Max(a,b) = a dacă a ≥ b şi Max(a,b) = b dacă a < b. Dacă se foloseşte simbolul ∨ pentru disjuncţie în locul Max, relaţia (9.7) se poate scrie

$$\mu_{A \cup B} = \mu_A \vee \mu_B \qquad (8.8)$$

Se poate arăta că reuniunea şi intersecţia sunt legate între ele prin identitatea

$$A \cup B = (A' \cap B')' \qquad (8.9)$$

unde A' şi B' reprezintă complementul mulţimilor fuzzy A şi B.

Formulele generale ale reuniunii şi intersecţiei sunt reprezentate de conormele triunghiulare (conormele T sau normele S) şi, respectiv de normele triunghiulare (normele T).

O normă T este o funcţie definită pe [0,1] × [0,1] cu valori în [0,1], care satisface criteriile:

1. $T(a,1) = a$
2. $T(a, b) \le T(c, d)$ dacă $a \le c,\ b \le d$
3. $T(a, b) = T(b, a)$
4. $T(T(a, b), c) = T(a, T(b, c))$

O normă S trebuie să îndeplinească următoarele condiţii:

1. $S(a, 0) = a$
2. $S(a, b) \le S(c, d)$, dacă $a \le c,\ b \le d$
3. $S(a, b) = S(b, a)$
4. $S(S(a, b),c) = S(a, S(b, c))$.

Complementul. Se spune că mulţimea fuzzy A' este complementul mulţimii fuzzy A, dacă şi numai dacă

$$\mu_{A'} = 1 - \mu_A \qquad (8.10)$$

Negația este interpretată ca fiind operația care înlocuieşte $\mu_A(x)$ prin $1 - \mu_A(x)$ pentru fiecare *x* din universul X.

Egalitatea. Două mulțimi fuzzy egale, A = B, dacă şi numai dacă $\mu_A(x) = \mu_B(x)$ pentru orice *x* din universul X.

Incluziunea. O mulțime fuzzy A este inclusă în mulțimea fuzzy B, sau este o submulțime a mulțimii fuzzy B (notație $A \subset B$), dacă şi numai dacă $\mu_A(x) \leq \mu_B(x)$.

Produsul algebric al mulțimilor fuzzy A şi B se notează AB şi se defineşte prin

$$\mu_{AB}(x) = \mu_A(x) \times \mu_B(x), \qquad x \in X \qquad (8.11)$$

Suma algebrică a mulțimilor fuzzy A şi B se notează $A \oplus B$ şi se defineşte prin

$$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \times \mu_B(x), \qquad x \in X \qquad (8.12)$$

Se poate arăta că

$$A \oplus B = (A' B')' \qquad (8.13)$$

- ***Relații fuzzy.***

Mulțimile fuzzy pot fi extinse astfel încât funcțiile de apartenență să aibă mai multe variabile. Aceste mulțimi fuzzy multi - dimensionale sunt numite în general *relații fuzzy*. O relație fuzzy *n*-ară R definită în $X_1 \times X_2 \times ... \times X_n$ este o mulțime fuzzy multidimensională definită pe $X_1 \times X_2 \times ... \times X_n$ şi notată

$$R = \{\mu_R(x_1,...,x_n) / (x_1,.....,x_n) / x_1 \in X_1,..., x_n \in X_n\} \qquad (8.14)$$

sau

$$R = \int_{X_1} ... \int_{X_n} \mu_R(x_1,...,x_n) / (x_1,...,x_n) = \int_{X_1} \mu_R(x_1,...,x_n) / (x_1,...,x_n) \qquad (8.15)$$

O astfel de relație poate reprezenta o corelație dintre elementele mulțimilor care definesc spațiul intrărilor şi ieşirii.

Relațiile fuzzy pot exprima asocieri lingvistice, corelații, relații, cum ar fi de exemplu: *mai mic decât*, *cam de două ori mai mare, mult mai mare*, etc.

Proprietățile mulțimilor fuzzy se pot extinde şi asupra relațiilor fuzzy, considerate ca mulțimi fuzzy multidimensionale.

- ***Proiecția si extensia cilindrica.***

Lotfi Zadeh, în anul 1975, a definit **proiecția** unei relații fuzzy în $X^i = X_{i_1} \times X_{i_2} \times \cdots \times X_{i_k}$ prin

$$proj\ (R; X^i) = \int_{X^i} \sup_{x_{j1}, \ldots, x_{jl}} \mu_R(x_{i_1}, \ldots, x_{i_k}) / (x_{i_1}, \ldots, x_{i_k}) \qquad (8.16)$$

în care R este o submulțime fuzzy pe $X^n = X_1 \times X_2 \times \ldots \times X_n$ şi $X^i \times X^j = X^n$. Indicii $j_1, \ldots, j_n$ sunt complementarii indicilor $i_1, \ldots, i_n$ față de indicii 1,2, ...,n. Conform acestei definiții, prin proiecție se elimină dimensiunile spațiului relației fuzzy prin luarea valorilor superioare ale funcției de apartenență pentru dimensiunea sau dimensiunile eliminate.

Extensia cilindrică a fost definită de Zadeh în 1975 astfel:

$$cext\ (R\ ;\ X^n) = \int_{X^n} \mu_R\ (x_{i_1}, \ldots, x_{i_k}) / (x_{i_1}, \ldots, x_{i_k}) \qquad (8.17)$$

unde R este o relație fuzzy în X^i. Aceasta înseamnă că o mulțime sau relație fuzzy este extinsă într-un spațiu cartezian, cu restricția că dacă R este o mulțime fuzzy X^n şi $X^n \subset X^m$, atunci:

$$R = proj\ (cext\ (R; X^m); X^m) \qquad (8.18)$$

Proiecția şi extensia cilindrică sunt utilizate în stabilirea relațiilor sau regulilor de compoziție.

- ***Relații de compoziție.***

Relația de compoziție a fost definită de Zadeh (1975) astfel: fie o relație R în $X \times Y$ şi A o mulțime fuzzy în X. O mulțime fuzzy B din Y poate fi dedusă din A, dată fiind relația de compoziție R şi mulțimea fuzzy A.

Aceasta se notează:

$$B = A \circ R \qquad (8.19)$$

şi este definită de :

$$B = proj\ (R \cap cext\ (A;\ X \times Y)) \qquad (8.20)$$

Considerând extensia cilindrică ca implicită, relația de compoziție poate fi interpretată ca având două faze: *intersecția de proiecție*.

Zadeh a propus folosirea compoziției *sup-min*. Dacă A este o mulțime fuzzy cu funcția de apartenență μ_A (x) şi R este o relație fuzzy cu funcția de apartenență $\mu_A(x,y)$, atunci relația de compoziție devine:

$$\mu_B\ (y) = \sup_x min\ (\mu_A(x),\ \mu_R\ (x,y)) \qquad (8.21)$$

unde extensia cilindrică este implicită iar *sup* şi *min* reprezintă fazele de proiecție şi respectiv combinare.

Alți cercetători au arătat că definiția relației de compoziție se poate generaliza prin considerarea normelor T şi S pentru operatorii *min* şi *sup*. De fapt, în domeniul continuu nu se poate considera o normă S (sau conormă T) generală deoarece modul ei de lucru nu este definit. De aceea, reprezentarea generală a unei relații fuzzy este:

$$\mu_B(y) = \sup_x T\ (\mu_A(x),\ \mu_R\ \mu_R\ (x,\ y)) \qquad (8.22)$$

unde T este norma T generală.

Implementarea *sup-min* propusă de Zadeh este cea mai utilizată.

8.1.2. Logica fuzzy.

Operațiile cu mulțimi fuzzy stau la baza operațiilor din logica fuzzy.

Propoziții de forma: *x este A*, unde *x* este o variabilă iar A este o valoare lingvistică, pot fi scrise în formalismul logicii fuzzy.

Se pot forma reguli şi baze de reguli care să fie folosite pentru procese de interferență logică.

- ***Propoziții fuzzy.***

Propozițiile fuzzy sunt de forma "*x este mai mare*", unde *mare* este o valoare lingvistică definită printr-o mulțime fuzzy pe universul discursului variabilei *x*. Propozițiile fuzzy sunt baza logicii fuzzy. Ele pot fi schimbate prin intermediul operatorilor logici. cum ar fi: operatorul *şi*, operatorul *sau*.

Modificatorii lingvistici pot fi folosiți pentru a modifica valorile lingvistice. De exemplu, modificatorul lingvistic *foarte* poate fi folosit pentru a modifica propoziția "*x este mare*" în "*x este foarte mare*".

Ca şi logica clasică, propozițiile fuzzy pot fi combinate cu ajutorul operatorilor *şi sau*. În logica fuzzy, aceşti operatori sunt implementați cu norme T şi respectiv, conorme T.

Operatorii cel mai des folosiți sunt operatorii propuşi de Zadeh:

ŞI min (a, b) SAU max (a, b)

Combinarea a două propoziţii fuzzy egale va avea ca rezultat aceiaşi informaţie după combinare:

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_A(x)) = \mu_A(x) \qquad (8.23)$$

$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_A(x)) = \mu_A(x)$$

În figura Fig. 8.1 sunt prezentate rezultatele aplicării operatorilor ŞI şi SAU propuşi de Zadeh:

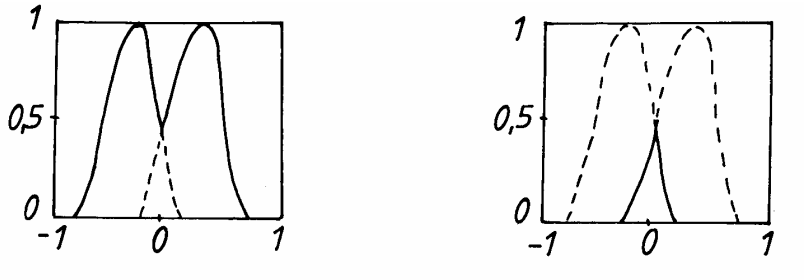


Fig. 8.1. Rezultatele aplicării operatorilor ŞI, SAU

Dacă propoziţiile se definesc pe universuri diferite, atunci aplicarea unui operator fuzzy va avea ca rezultat o relaţie fuzzy.

De exemplu, propoziţia:

p: x_1 este A_1 ***şi*** x_2 este A_2

poate fi reprezentată de relaţia fuzzy:

$$\mu_P(x_1,x_2) = T(\mu_{A1}(x_1), \mu_{A2}(x_2)) \qquad (8.24)$$

în care T este o normă T generală care modelează operatorul ***şi***. O asemenea combinaţie de propoziţii poate constitui premisa unei reguli fuzzy.

- ***Reguli fuzzy.***

Regulile fuzzy se reprezintă prin implicanţi fuzzy, care au aceleaşi rol ca şi tabelul de adevăr din implicaţia definită în logica clasică. În logica clasică, propoziţia: **dacă A atunci B** este modelată prin implicaţia logică $\mathbf{A \rightarrow B}$.

O regulă fuzzy poate fi reprezentată printr-o relaţie fuzzy cu ajutorul unui implicant fuzzy. Regulile fuzzy se pot combina prin operatori de asigurare.

Regula fuzzy este de forma *dacă-atunci*, unde premisa şi consecinţa regulii sunt propoziţii fuzzy. Premisa poate conţine o combinaţie de propoziţii realizată cu operatorii *şi, sau* ori cu operatorii de negare. O propoziţie de forma

dacă x_1 *este* A_1 **şi** x_2 *este* A_2, ***atunci*** *y este B* se poate reprezenta sub forma relaţiei:

$$R = I(T(A_1, A_2), B) \qquad (8.25)$$

unde T este o conjuncţie bazată pe o normă T generală şi I este un implicant fuzzy. Norma T modelează operatorul *şi* iar implicantul fuzzy I modelează implicaţia *dacă-atunci*. Rezultă că o regulă fuzzy se poate reprezenta printr-o relaţie fuzzy. Funcţia de apartenenţă a relaţiei R este:

$$\mu_R(x_1, x_2, y) = I[T(\mu_{A1}(x_1), \mu_{A2}(x_2)), \mu_B(y)] \qquad (8.26)$$

Dacă există mai mult de o propoziţie în consecinţa unei reguli fuzzy, atunci se admite că regulile fuzzy sunt separabile, obţinându-se un set de reguli cu o singură consecinţă. Acest set de reguli, care au o singură consecinţă, se combină într-o relaţie fuzzy. Mai multe reguli formează o **bază de reguli**. Regulile fuzzy sunt considerate ca un set de *r* reguli paralele care au premisele bazate pe *n* variabile:

r_1: **dacă** x_1 *este* A_1 **şi** ... **şi** x_m *este* A_m **atunci** *y este* B_1
altfel ...

altfel
r_K: **dacă** x_K *este* A_K **şi** ... **şi** x_n *este* A_n **atunci** *y este* B_K
altfel ...

altfel
dacă x_r *este* A_r **şi** ... **şi** x_n *este* A_n **atunci** y este B_r

Modelarea unui astfel de set de reguli paralele cu o relaţie fuzzy se efectuează prin construirea relaţiei R_K pentru fiecare regulă r_K şi combinarea acestor relaţii într-o singură relaţie R. Această combinare a relaţiilor fuzzy într-o singură relaţie se numeşte **agreare**.

- ***Inferenţa fuzzy.***

Inferenţa se referă la o regulă fuzzy sau la o bază de reguli. Inferenţa unei reguli fuzzy este o aplicare directă a compunerii relaţiilor fuzzy. Acest tip de inferenţă, introdus de Zadeh, presupune că o regulă fuzzy de tipul:

dacă *x este A* **atunci** *y este B*

este reprezentată de o relaţie fuzzy R.

În cazul în care relaţia R este dată, se deosebesc următoarele tipuri de inferenţă:

a) fiind dată premisa A' se poate infera consecinţa B';

b) fiind dată consecinţa B' se poate obţine premisa A'.

De obicei se utilizează legea de compoziţie *sup-min*, deşi sunt posibile şi alte combinaţii.

Conform tipului de inferenţă *a),* inferenţa se realizează prin *regula de compoziţie* pentru inferenţă

$$B' = A' \circ R \qquad (8.27)$$

în care R reprezintă regula fuzzy: **dacă** *x este A* **atunci** *y este B*.

Pentru inferenţa de tipul *b)* se poate folosi *regula de compoziţie* pentru interferenţa fuzzy

$$A' = R \circ B' \qquad (8.28)$$

Consecinţele sunt reprezentate printr-un set de reguli fuzzy paralele, numit **bază de reguli fuzzy**. Agrearea regulilor fuzzy se efectuează diferit în funcţie de tipul implicaţiilor.

8.1.3. Sisteme de conducere automată bazate pe logica fuzzy.

Conducerea fuzzy este cea mai dezvoltată aplicaţie a teoriei mulţimilor fuzzy. Un algoritm fuzzy de control reprezintă de fapt aplicarea regulii de compoziţie pentru inferenţă. Fiind dată o relaţie R reprezentând baza de reguli pentru conducere şi intrările fuzzy A', se poate obţine o ieşire fuzzy B' prin compunerea lui A' cu R, adică $B' = A' \circ R$. Algoritmul de control este descris prin reguli fuzzy în care premisele sunt clasificări ale intrărilor iar consecinţele sunt clasificări ale incrementului comenzii. S-a precizat că regulile fuzzy pot fi reprezentate prin relaţii fuzzy şi că relaţiile se pot agrea.

La sistemele de conducere automată intrările şi ieşirile sunt mărimi numerice, deci este necesară o transformare a intrărilor numerice în intrări fuzzy, operaţie numită **fuzificare** (vaguizare) şi o transformare a ieşirilor fuzzy în valori numerice, operaţie numită **defuzificare** (devaguizare).

În figura Fig. 8.2 este prezentată o schemă bloc a algoritmilor de conducere fuzzy.

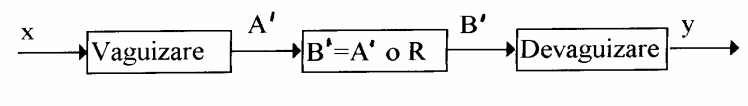


Fig. 8.2 Schema bloc a algoritmilor de conducere fuzzy

Dacă intrarea este o valoare numerică, adică nu are nici un grad de incertitudine (cum este cazul mărimii măsurate pe calea de reacţie sau referinţă) atunci se spune că mulţimea fuzzy A' este un *singleton*:

$$\begin{cases} 1 \text{ dacă } x_i = x_i' \end{cases}$$

$$\mu_{Ai}(x_i) = \begin{cases} \\ \\ 0 \text{ altfel} \end{cases} \qquad (8.29)$$

Defuzzyficarea unei relații fuzzy se poate efectua prin metodele *centrului* de masă, *media maximilor* sau *centrul ariilor*.

Metoda *centrului de masă* este similară cu calculul centrului de masă al unui corp şi este metoda cea mai folosită. Diferența este că se înlocuieşte masa corpului cu valorile funcției de apartenență:

$$\mathrm{cm}(B') = \frac{\int_Y \mu_{B'}(y)\, y dy}{\int_Y \mu_{B'}(y) dy} \qquad (8.30)$$

sau, în domeniul discret:

$$\mathrm{cm}(B') = \frac{\sum_{i=1}^{m} \mu_{B'}(y_i)\, y_i}{\sum_{i=1}^{m} \mu_{B'}(y_i)} \qquad (8.31)$$

Regulile de conducere fuzzy pentru un regulator fuzzy cu o intrare şi o ieşire pot fi scrise, spre exemplu, astfel:

dacă *E este PP, VE este MP*, **atunci** *VO este NN*;
dacă *E este AZ, VE este AZ*, **atunci** *VO este AZ*,

unde:

E - eroarea (abaterea);
VE - variația erorii;
VO - variația ieşirii;
PP - puțin pozitivă;
MP - mult pozitivă;
NN - normal negativă;
AZ - aproape zero.

Este necesară determinarea mulțimii de bază sau a universului pentru mărimile fuzzy. Universul în conducerea fuzzy poate fi:

- discretizat, spre exemplu între − 6 şi + 6;
- continuu, în intervalul [−1, 1].

Univers discretizat. În acest caz, variabilele fuzzy sunt alese uzual aşa cum se arată în tabelul 9.1. Este alegerea grupului de cercetători conduşi de Mamdani de la Queen Mary College.

Tab 8.2

	- 6	5	4	3	2	1	0	1	2	3	4	5	6
MP													0
NP											0		
PP									0				
AZ							0						
PN					0								
NN			0										
MN	10	7											

MP - mult pozitiv; NP - normal pozitiv; PP - puțin pozitiv; AZ - aproape zero; PN - puțin negativ; NN - normal negativ; MN - mult negativ.

Din tabel se observă că fiecare variabilă fuzzy A se exprimă într-un mod, conform relației următoare:

$$A = \frac{a_1}{x_1} + \frac{a_2}{x_2} + \frac{1}{x_3} + \frac{a_2}{x_4} + \frac{a_1}{x_5} \qquad (8.32)$$

care este caracterizată de trei parametri a_1, a_2 şi x_3 care are cel mai mare grad de apartenență, adică 1.

Univers continuu. Variabilele fuzzy sunt alese uzual aşa cum se arată în Fig. 8.3 şi Fig. 8.4.

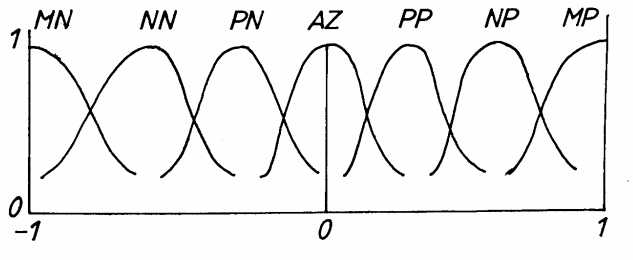


Fig. 8.3 Variabilele fuzzy de tip exponențial

Pentru variabilele fuzzy de tip exponențial se pot asocia funcții de apartenență cum ar fi de exemplu:

$$A(x) = \exp[-a^2(x - x_0)] \qquad (8.33)$$

care este caracterizată de doi parametri, *a* şi x_0.

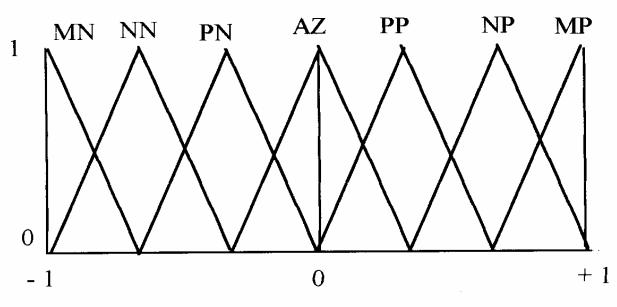


Fig. 8.4 Variabile fuzzy de tip triunghiular

Funcţia de apartenenţă de tip triunghiular Fig. 8.4 este caracterizată de asemenea de doi parametri.

Există şi un alt tip de variabile fuzzy, cu funcţia de apartenenţă monotonă , aşa cum este în Fig. 8.5.

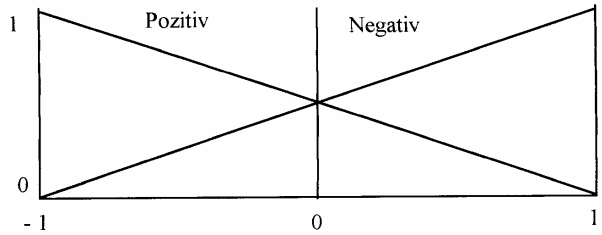


Fig. 8.5 Variabile fuzzy de tip monoton

În general, tipul unei variabile fuzzy este în aceeaşi măsură legat de *metoda de raţionament fuzzy.*

Există în general două metode de raţionament fuzzy:

1. metoda bazată pe reguli de compoziţie a interferenţei;
2. metoda bazată pe logica fuzzy.

În conducerea fuzzy, metodele convenţionale de raţionament fuzzy sunt folosite frecvent datorită simplităţii calculelor.

- ***Metoda de raţionament fuzzy bazata pe reguli de compoziţie a interferenţei.***

Fie un regulator fuzzy cu intrările x_1 şi x_2 şi ieşirea *y*. Să presupunem că avem următoarele două implicaţii:

dacă x_1 *este* A_{11}, x_2 este A_{12}, **atunci** *y este* B_1;
dacă x_1 *este* A_{21}, x_2 este A_{22}, **atunci** *y este* B_2.

Atunci, dându-se x_1° şi x_2°, valorile ponderilor μ_1 şi μ_2 ale premiselor se calculează fiecare conform cu relaţiile:

$$\mu_1 = A_{11}(x_1^\circ) \wedge A_{12}(x_2^\circ)$$

$$\mu_2 = A_{21}(x_1^\circ) \wedge A_{22}(x_2^\circ) \qquad (8.34)$$

sau ca relațiile:

$$\mu_1 = A_{11}(x_1^\circ) \times A_{12}(x_2^\circ)$$

$$\mu_2 = A_{21}(x_1^\circ) \times A_{22}(x_2^\circ) \qquad (8.35)$$

Prima implicație sugerează $\mu_1 B_1$ şi a doua implicație sugerează $\mu_2 B_2$, unde:

$$(\mu_i B_i)(y) = \mu_i \times B_i(y) \qquad i = 1,2 \qquad (8.36)$$

Atunci se formează o mulțime fuzzy B^* astfel:

$$B^* = \mu_1 B_1 \cup \mu_2 B_2 \qquad (8.37)$$

Ieşirea y° se deduce cu ajutorul *centrului de masă* al lui $B^*(y)$:

$$y^\circ = \frac{\int B^*(y)\, y\, dy}{\int B^*(y)\, dy} \qquad (8.38)$$

Raționamentul este ilustrat în Fig. 8.6.

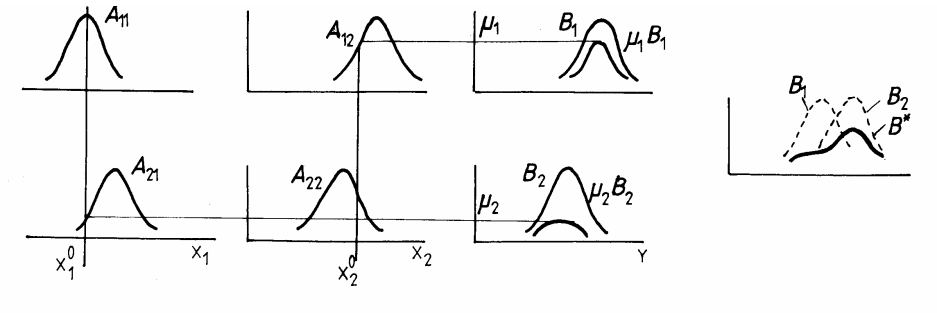


Fig. 8.6 Raţionament fuzzy bazat pe reguli de compoziţie a interferenţei

Dacă există mai multe reguli de conducere aplicabile la intrare, atunci regulile a căror importanţă este mai mică se omit din raţionament. Metoda de agreare a ieşirilor deduse din mai multe reguli a fost cândva o problemă mult discutată. Acum, cele mai multe regulatoare fuzzy adoptă media ponderată. Un regulator fuzzy care utilizează acest tip de raţionament se spune că "funcţionează ca un multireleu".

- ***Metoda de raţionament bazata pe logica fuzzy.***

Este o metodă bazată pe logică fuzzy, în care se utilizează variabile fuzzy cu funcţia de apartenenţă monotonă. Să considerăm următoarele două implicaţii:

dacă x_1 *este N,* x_2 *este P*, **atunci** *y este N*;
dacă x_1 *este P,* x_2 *este N*, **atunci** *y este P*.

în care *P* = pozitiv, *N* = negativ.
Raţionamentul este ilustrat în Fig. 8.7.

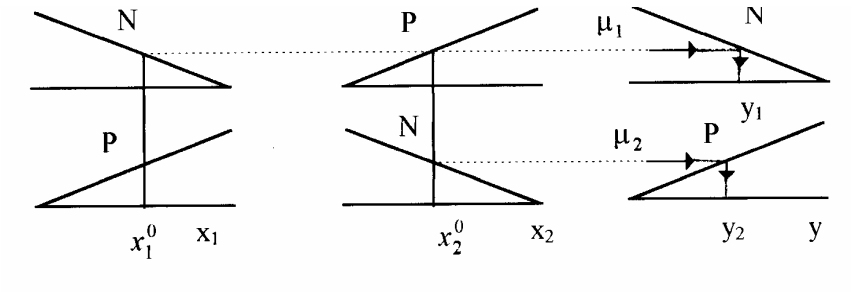


Fig. 8.7 Raţionament bazat pe logica fuzzy

Fie

$$\mu_1 = N\ (y_1)$$

$$\mu_2 = P\ (y_2) \qquad (8.39)$$

atunci pentru o intrare x_1°, x_2°, ieşirea y° se deduce cu relaţia:

$$y^\circ = \frac{\mu_1 y_1 + \mu_2 y_2}{\mu_1 + \mu_2} \qquad (8.40)$$

în care μ_1 şi μ_2 sunt ponderile primei şi celei de-a doua implicaţii.

- Referitor la prima metodă de raţionament, apar dificultăţi atunci când numărul intrărilor devine mare. Frecvent se utilizează cinci variabile pentru fiecare intrare x_i . Dacă există n intrări, numărul regulilor de conducere devine 5^n, deci este dificilă administrarea lor pentru n mare. Regulile de conducere fuzzy de acest tip exprimă o relaţie numai între cantităţi fuzzy.
- Atunci când se utilizează a doua metodă de raţionament, se poate reduce numărul regulilor necesare pentru conducere. Spre exemplu, o relaţie liniară între x_1, x_2 şi y poate fi exprimată cu ajutorul a patru reguli. Totuşi, o regulă de conducere este de fapt o relaţie între cantităţi fuzzy, iar regulamentul exprimă o valoare fuzzy a ieşirii y.
- Referitor la premise, este interesant de observat ce înţelegem de obicei prin propoziţia fuzzy " *x_1 este mai mare, x_2 este mai mic*" în spaţiul fuzzy x_1 - x_2.

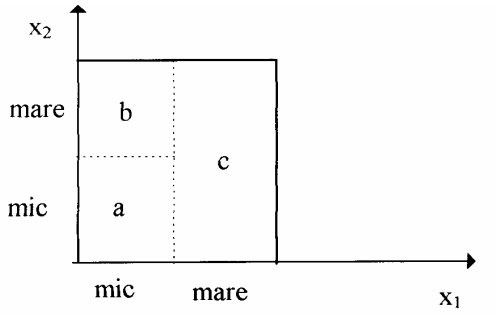


Fig. 8.8. Separare fuzzy a spaţiului intrărilor

Premisele unui set de reguli descriu o separare fuzzy a spaţiului intrărilor aşa cum se arată în Fig. 8.8. Subspaţiului *(a)* îi corespunde propoziţia fuzzy " x_1

este mic şi x_2 este mic", subspaţiului *(b)* îi corespunde " x_1 este mic şi x_2 este mare" şi subspaţiului *(c)* îi corespunde numai " x_1 este mare".

- Se poate considera că există şi a treia metodă de raţionament fuzzy, atunci când avem următoarele implicaţii:

  dacă x_1 *este* A_{11}, x_2 *este* A_{12}, **atunci** $y = f_1(x_1, x_2)$;

  dacă x_1 *este* A_{21}, x_2 *este* A_{22}, **atunci** $y = f_2(x_1, x_2)$.

Pentru x_1° şi x_2°, y° se deduce relaţia:

$$y^\circ = \frac{\mu_1 f_1(x_1^0, x_2^0) + \mu_2 f_2(x_1^0, x_2^0)}{\mu_1 + \mu_2} \qquad (8.41)$$

unde μ_i reprezintă ponderea implicaţiei *i*.

- ***Proiectarea regulatoarelor fuzzy***

Cel mai important aspect legat de conducerea fuzzy este modul de operare al regulatorului fuzzy, prin care să se obţină regulile de conducere fuzzy. Se cunosc până în prezent mai multe metode de proiectare, dintre care una va fi prezentate în continuare cu avantajele şi dezavantajele ei.

Numeroase regulatoare fuzzy au fost proiectate până acum folosind experienţa operatorului uman şi / sau cunoştinţele din domeniul conducerii automate. Se poate afirma că această conducere fuzzy a fost prima aplicaţie practică a sistemelor expert.

În mai multe cazuri, în care operatorul uman deţine un rol important în conducerea procesului, sunt uşor de obţinut cunoştinţele sale legate de conducere prin intervievare şi transformarea lor în implicaţii fuzzy. Este de asemenea posibil pentru un inginer automatist să elaboreze un număr mare de protocoale bazate pe cunoştinţele sale despre procesul automatizat şi pe cunoştinţele generale de conducere a proceselor.

Spre exemplu, să presupunem un proces automatizat şi răspunsul indicial al sistemului automat (Fig. 8.9). Răspunsul indicial este divizat în patru faze I, II, III, IV, astfel încât este uşor să se analizeze acţiunea ieşirii în anumite puncte ale răspunsului în scopul îmbunătăţirii performanţelor.

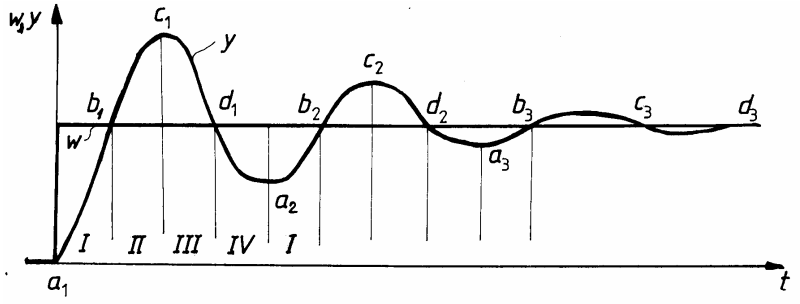


Fig. 8.9 Răspunsul sistemului automat cu regulator clasic

Regulile de conducere primare pot fi stabilite conform tabelului 9.2.

Tab 8.3

Nr. regulă	Eroare (E)	Variația erorii (VE)	Variația ieşirii (VO)	Punctul de pe răspunsul indicial
1	MP	AZ	MP	a_1
2	AZ	MP	MN	b_1
3	MN	AZ	MN	c_1
4	AZ	MN	MP	d_1
5	NP	AZ	NP	a_2
6	AZ	NP	NN	b_2
7	NN	AZ	NN	c_2
8	AZ	NN	NP	d_2
9	PP	AZ	PP	a_3
10	AZ	PP	PN	b_3
11	PN	AZ	PN	c_3
12	AZ	PN	PP	d_3
13	AZ	AZ	AZ	w = y

În tabel s-au folosit următoarele notații: MP - mult pozitivă, AZ - aproape zero, MN - mult negativă, NP - normal pozitivă, NN - normal negativă, PP - puțin pozitivă, PN - puțin negativă.

Prima regulă de conducere corespunde punctului de start a_1 din faza I, care este descrisă astfel: " *E este MP şi VE este AZ*". A doua regulă corespunde punctului b_1 şi aşa mai departe. Cea de-a treisprezecea regulă corespunde stabilizării sistemului la valoarea de referinţă.

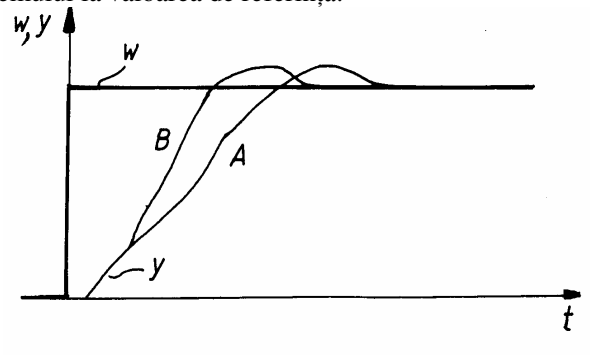


Fig. 8.10 Răspunsul sistemului automat cu regulator fuzzy

Regulatorul fuzzy implementat pe baza acestor reguli conduce la răspunsul A al sistemului automat ca în Fig. 8.10. Răspunsul acesta poate fi îmbunătățit prin adăugarea mai multor reguli de conducere fuzzy. Spre exemplu, putem considera o regulă într-un nou punct al răspunsului, ceva mai sus de punctul a_1:

dacă *E este MP, VE este PP*, **atunci** *VO este NP*

pentru a micşora timpul de creştere (performanţă a sistemului automat). Este recomandat, de asemenea, să se considere mai jos (mai înainte) de punctul b_1 următoarea regulă de conducere fuzzy:

dacă *E este PP, VE este MP*, **atunci** *VO este NN*

pentru a reduce suprareglarea, în condiţiile în care procesul este caracterizat de o anumită constantă de timp (inerţie).

În această manieră se pot stabili şase reguli noi de conducere fuzzy (14 - 19) care se adaugă la cele stabilite anterior (vezi tabelul 9.3).

Tab 8.4

Nr. regulă	Eroarea (E)	Variaţia erorii (VE)	Variaţia ieşirii (VO)	Punctul de pe răspunsul indicial
14	MP	PP	NP	a_1 - b_1
15	PP	MP	NN	a_1 - b_1
16	MN	PN	NN	c_1 - d_1
17	PN	MN	NP	c_1 - d_1
18	PP	PP	AZ	a_3 - b_3
19	PN	PN	AZ	c_3 - d_3

Cu ajutorul acestor 19 reguli de conducere fuzzy se obţine răspunsul indicial B din figura Fig. 8.10, îmbunătăţit faţă de răspunsul A.

Metoda de proiectare a regulatorului fuzzy prezentată, are câteva dezavantaje:

a) inginerul automatist obţine cu dificultate informaţiile necesare de la operatorul uman care, de obicei, poate mai uşor să facă decât să explice ceea ce face;
b) dacă procesul este complex, este dificilă scrierea regulilor de conducere fuzzy în sensul cerut de conducerea automată;
c) este dificilă stabilirea unei proceduri generale de proiectare a regulatoarelor fuzzy, deoarece metoda se bazează pe principii euristice.

8.1.4. Avantajele conducerii fuzzy si domenii de aplicare.

Ideea de conducere fuzzy se caracterizează printr-o strategie de conducere exprimată printr-un număr de reguli de conducere fuzzy.

Aşa cum s-a arătat, o regulă de conducere fuzzy este exprimată printr-o implicaţie fuzzy de forma *" dacă ... atunci...",* care include variabile fuzzy, numite adesea variabile lingvistice. Implicaţiile fuzzy, precum şi mulţimile fuzzy sunt foarte potrivite în descrierea procesului supus automatizării cu ajutorul felului de a gândi al omului. Aceasta deoarece propoziţiile fuzzy în implicaţiile fuzzy sunt mai degrabă calitative decât cantitative. Se poate spune că gândirea umană este *calitativă*, bazată pe termeni lingvistici şi foloseşte adesea informaţii *parţiale* în situaţii locale. Aceste două trăsături, calitativ şi parţial sunt esenţiale şi pot fi exprimate cu ajutorul mulţimilor fuzzy şi al implicaţiilor fuzzy: o implicaţie fuzzy descrie numai cunoaşterea parţială sau informaţia într-o situaţie locală.

În continuare să ne concentrăm atenția asupra situației conducerii unui proces de către un operator uman. Un operator experimentat ia decizii în situații specifice bazându-se pe cunoaşterea imperfectă şi parțială, care a fost acumulată datorită lungii sale experiențe în conducerea procesului. Este posibil uneori ca decizia să reia în condițiile în care informația disponibilă este fuzzy sau limitată. Deci există circumstanțe favorabile introducerii ideii de conducere fuzzy.

Totuşi, nu este suficientă pentru conducere exprimarea calitativă a unei situații sau a unui protocol. Este necesară executarea unei idei calitative într-o situație reală. Mulțimile fuzzy sunt potrivite acestui scop, deoarece executarea unui set de reguli de conducere fuzzy este îndeplinită numeric printr-o metodă de raționament fuzzy, în care agrearea ieşirilor deduse din diferite reguli este bine îndeplinită.

Prin utilizarea implicațiilor fuzzy, proiectantul poate îngloba într-un regulator o varietate de idei utile. Odată proiectat, un regulator fuzzy lucrează ca şi un regulator obişnuit, în care s-au implementat relații funcționale între intrări şi ieşiri.

Teoria logicii fuzzy este utilizată cel mai mult în domeniul produselor de larg consum, cum ar fi: maşinile de spălat, televizoarele color, camerele video, aparatele de fotografiat etc. Industria automobilelor este a doua, după industria bunurilor de consum, care aplică masiv tehnologii bazate pe logica fuzzy: controlul transmisiei, al funcționării motorului sau al sistemului antiblocare al frânelor. În domeniul conducerii automate a proceselor există aplicații în industriile aluminiului, chimică, sticlei, metalurgică.

În prezent există numeroase firme importante producătoare de echipamente de automatizare care folosesc logica fuzzy. Este remarcabil faptul că logica fuzzy a pătruns şi pe piața automatelor programabile. Astfel, compania americană Allen-Bredley oferă un software (A-B Flex) care permite simularea pe calculator şi apoi implementarea pe automate programabile a algoritmilor fuzzy de control. Firma germană Klöckner-Moeller produce automatul programabil SUCOcontrol PS-401, care evaluează o regulă fuzzy în mai puțin de 1m/sec. Programarea automatului se efectuează cu sistemul *fuzzy TECH* al firmei Inform Software Corp. Concernul Siemens are de asemenea preocupări în domeniul conducerii fuzzy aplicate în proiecte specifice de automatizare, producerea şi transportul energiei, sisteme de semnalizare pentru căi ferate.

Instalațiile din clădiri pot fi conduse cu succes pe baza logicii fuzzy şi în această direcție se cunosc realizări ale conducerii fuzzy pentru instalații de încălzire, ventilare şi climatizare.

8.1.5. Sinteza algoritmului de conducere fuzzy pentru o instalație.

Se va prezenta o aplicație care utilizează teoria mulțimilor fuzzy în scopul reglării automate a temperaturii într-o instalație de încălzire.

Se defineşte *universul* sau *mulțimea de bază* pentru temperatură astfel:

$$T = \left\{ \frac{1}{t_1}, \frac{1}{t_2}, \frac{1}{t_3} \right\} \qquad (8.42)$$

Cu ajutorul operației de fuzzyficare se definesc submulțimile fuzzy:

$$temperatură\ mică = \left\{ \frac{1}{t_1}, \frac{0,4}{t_2}, \frac{0,1}{t_3} \right\} \tag{8.43}$$

$$temperatură\ mare = \left\{ \frac{0,1}{t_1}, \frac{0,4}{t_2}, \frac{1}{t_3} \right\} \tag{8.44}$$

după modelul submulțimii A

$$A = \left\{ \frac{\mu_A(t_1)}{t_1}, \frac{\mu_A(t_2)}{t_2}, \frac{\mu_A(t_3)}{t_3} \right\} \tag{8.45}$$

în care $\mu_A(t_1)$ este coeficientul de apartenență al lui t_1 la submulțimea A

$$\mu_A(t) \in [0,1] \tag{8.46}$$

Conform operației de intersecție a mulțimilor, pornind de la submulțimile fuzzy (9.43) şi (9.44) se poate defini submulțimea fuzzy "temperatură mică şi mare" astfel:

$$temperatură\ mică\ şi\ mare = \left\{ \frac{0,1}{t_1}, \frac{0,4}{t_2}, \frac{0,1}{t_3} \right\} \tag{8.47}$$

Se defineşte *universul tensiunii de comandă* a elementului de execuție

$$U = \left\{ \frac{1}{u_1}, \frac{1}{u_2}, \frac{1}{u_3} \right\} \tag{8.48}$$

în care u_1=2V, u_2=5V, u_3=8V şi submulțimile fuzzy

$$tensiune\ mică = \left\{ \frac{0,9}{u_1}, \frac{0,2}{u_2}, \frac{0}{u_3} \right\} \tag{8.49}$$

$$tensiune\ moderată = \left\{ \frac{0,3}{u_1}, \frac{0,9}{u_2}, \frac{0,3}{u_3} \right\} \tag{8.50}$$

$$tensiune\ mare = \left\{ \frac{0}{u_1}, \frac{0,2}{u_2}, \frac{0,9}{u_3} \right\} \tag{8.51}$$

Între două submulțimi fuzzy A şi B, definite pe universuri diferite, se defineşte *implicația lingvistică (regula R) astfel:*

R: **dacă A atunci B**

sau cu altă notație

R: **A ----> B**

Relația de implicare R este determinată cu ajutorul produsului cartezian A x B astfel:

$$\mu_R(t,u) = \mu_{AxB}(t,u) = \min\left[\mu_A(t), \mu_B(u)\right], \quad t \in T, \quad u \in U \qquad (8.52)$$

Pentru reglarea temperaturii într-o instalație de încălzire se defineşte un *algoritm fuzzy de conducere* determinat de următoarele reguli care formează o *bază de reguli*:

R_1: temperatură mică $\longrightarrow$ tensiune mare

R_2: temperatură mare $\longrightarrow$ tensiune mică

R_3: temperatură mică $\cap$ mare $\longrightarrow$ tensiune moderată

Aceste trei reguli constituie un *algoritm fuzzy exprimat lingvistic.*

Regula R_1 (temperatură mică în instalație implică tensiune mare de comandă), pe baza relațiilor (9.43), (9.51) şi (9.52) conduce la:

$$R_1 = \left\{ \begin{matrix} \frac{0}{t_1,u_1} & \frac{0{,}2}{t_1,u_2} & \frac{0{,}9}{t_1,u_3} \\ \frac{0}{t_2,u_1} & \frac{0{,}2}{t_2,u_2} & \frac{0{,}4}{t_2,u_3} \\ \frac{0}{t_3,u_1} & \frac{0{,}1}{t_3,u_2} & \frac{0{,}1}{t_3,u_3} \end{matrix} \right\} \qquad (8.53)$$

Pentru a facilita înțelegerea modului de scriere al (9.39) se prezintă următorul exemplu:

$$\mu_{R_1}(t_3,u_2) = \min\left[\mu_{mică}(t_3), \mu_{mare}(u_2)\right] = \min\,[0{,}1,\ 0{,}2] = 0{,}1$$

Similar se scriu şi regulile R_2 şi R_3:

$$R_2 = \left\{ \begin{matrix} \frac{0{,}1}{t_1,u_1} & \frac{0{,}1}{t_1,u_2} & \frac{0}{t_1,u_3} \\ \frac{0{,}4}{t_2,u_1} & \frac{0{,}2}{t_2,u_2} & \frac{0}{t_2,u_3} \\ \frac{0{,}9}{t_3,u_1} & \frac{0{,}2}{t_3,u_2} & \frac{0}{t_3,u_3} \end{matrix} \right\} \qquad (8.54)$$

$$R_3 = \left\{ \begin{array}{ccc} \dfrac{0,1}{t_1,u_1} & \dfrac{0,1}{t_1,u_2} & \dfrac{0,1}{t_1,u_3} \\ \dfrac{0,3}{t_2,u_1} & \dfrac{0,1}{t_2,u_2} & \dfrac{0,3}{t_2,u_3} \\ \dfrac{0,1}{t_3,u_1} & \dfrac{0,1}{t_3,u_2} & \dfrac{0,1}{t_3,u_3} \end{array} \right\} \qquad (8.55)$$

Se procedează la agregarea celor trei reguli fuzzy într-o singură regulă:

$$R = R_1 \cup R_2 \cup R_3 = \left\{ \begin{array}{ccc} \dfrac{0,1}{t_1,u_1} & \dfrac{0,2}{t_1,u_2} & \dfrac{0,9}{t_1,u_3} \\ \dfrac{0,4}{t_2,u_1} & \dfrac{0,4}{t_2,u_2} & \dfrac{0,4}{t_2,u_3} \\ \dfrac{0,9}{t_3,u_1} & \dfrac{0,2}{t_3,u_2} & \dfrac{0,1}{t_3,u_3} \end{array} \right\} \qquad (8.56)$$

Dacă se cunoaşte algoritmul fuzzy de conducere şi o măsurare fuzzy T', se poate calcula comanda fuzzy U' cu ajutorul regulii de compoziţie a lui Zadeh (9.28) astfel:

$$U' = R \circ T' \qquad (8.57)$$

definită de prin

$$U' = proj\left[R \cap cext\left(T';T \times U\right)\right] \qquad (8.58)$$

Ţinând seama de funcţiile de apartenenţă, conform relaţiei (7.21) se poate scrie:

$$\mu_{U'} = \sup_t \min\left[\mu_{T'}(t), \mu_R(t,u)\right] \qquad (8.59)$$

Exemplu. Operatorul măsoară fuzzy următoarea temperatură:

$$T' = mic\ mediu = \left\{ \frac{0,5}{t_1}, \frac{0,4}{t_2}, \frac{0,2}{t_3} \right\} \qquad (8.60)$$

Se aplică relaţia (9.57) şi se obţine:

$$\left\{\frac{0,1}{t_1,u_1}\ \frac{0,2}{t_1,u_2}\ \frac{0,9}{t_1,u_3}\ \frac{0,4}{t_2,u_1}\ \frac{0,4}{t_2,u_2}\ \frac{0,3}{t_2,u_3}\ \frac{0,1}{t_3,u_1}\ \frac{0,1}{t_3,u_2}\ \frac{0,1}{t_3,u_3}\right\} \cap \left\{\frac{0,5}{t_1,u_1}\ \frac{0,5}{t_1,u_2}\ \frac{0,5}{t_1,u_3}\ \frac{0,4}{t_2,u_1}\ \frac{0,4}{t_2,u_2}\ \frac{0,4}{t_2,u_3}\ \frac{0,2}{t_3,u_1}\ \frac{0,2}{t_3,u_2}\ \frac{0,2}{t_3,u_3}\right\} \overset{\min}{=} \left\{\frac{0,1}{t_1,u_1}\ \frac{0,2}{t_1,u_2}\ \frac{0,5}{t_1,u_3}\ \frac{0,4}{t_2,u_1}\ \frac{0,4}{t_2,u_2}\ \frac{0,3}{t_2,u_3}\ \frac{0}{t_3,u_1}\ \frac{0,1}{t_3,u_2}\ \frac{0,1}{t_3,u_3}\right\}$$

$$\xrightarrow[\sup\limits_{t}]{proj} = \left\{\frac{0,4}{u_1}, \frac{0,4}{u_2}, \frac{0,5}{u_3}\right\}$$

S-a obținut comanda fuzzy

$$U' = \left\{\frac{0,4}{u_1}, \frac{0,4}{u_2}, \frac{0,5}{u_3}\right\} \tag{8.61}$$

care trebuie defuzzyficată, de exemplu prin metoda maximului.

$$u = u_3 = 8V \tag{8.62}$$

Dacă măsurarea este nefuzzy, de exemplu $t = t_1 = 10^0C$

$$T = \left\{\frac{1}{t_1}, \frac{0}{t_2}, \frac{0}{t_3}\right\} \tag{8.63}$$

atunci ieşirea U' se determină uşor prin căutarea în tabloul R a coeficienților μ_R corespunzător lui t_1:

$$U' = \left\{\frac{0,1}{u_1}, \frac{0,2}{u_2}, \frac{0,9}{u_3}\right\} \tag{8.64}$$

Regulator de tip Mamdani

Conform bazei de reguli care constituie algoritmul fuzzy de reglare a temperaturii în instalația de încălzire, coeficienții de apartenență ai temperaturii *t* şi ai comenzii *u* au variații aşa cum se prezintă în figura 15.11. Regula R se obține prin agregarea celor trei reguli R_1, R_2 şi R_3 iar coeficientul de apartenență corespunzător este dat de relația

$$\mu_R = \mu_{R_1 \cup R_2 \cup R_3} = \max\left(\mu_{R_1}, \mu_{R_2}, \mu_{R_3}\right) \tag{8.65}$$

Determinarea comenzii u_0 a regulatorului (defuzzyficarea) care să asigure temperatura t_0 în instalație se obține prin metoda centrului de masă astfel:

$$u_0 = \frac{\int \mu(u) du}{\int \mu(u) du} \tag{8.66}$$

Regulator de tip Sugeno

Calculul comenzii regulatorului de tip Sugeno se efectuează conform figurii 15.12. Coeficienţii de apartenenţă ai temperaturii *t* se modifică corespunzător regulilor R_1, R_2 şi R_3 identic ca şi în cazul regulatorului Mamdani. Deosebirile apar la stabilirea valorii comenzii regulatorului. Comanda u_0 *a* regulatorului se calculează cu relaţia:

$$u_0 = \frac{\mu_a \cdot 8 + \mu_b \cdot 2 + \mu_c \cdot 5}{\mu_a + \mu_b + \mu_c} \qquad (8.67)$$

8.1.6. Simularea sistemelor bazate pe logica fuzzy cu ajutorul KitSAS.

Kitul de simulare poseda un element cu mnemonicul **fuzzy** de tip Sugeno. Regulile care determina comportarea elementului sunt specificate într-un fişier a cărui nume, cu calea completă, este indicat este indicat în ultimul câmp Txt. al structurii sale de date. Ieşirea elementului este determinată după două formule:

- Dacă fişierul cu reguli conţine numai o singură regulă, atunci ieşirea este egală cu factorul de apartenenţă Miu corespunzător intrării înmulţit cu comanda U. Dacă se ia U egal cu 1 se obţine chiar funcţia Miu(X) pentru acea regulă.
- Dacă fişierul cu reguli conţine mai multe reguli atunci ieşirea este dată de relaţia lui Sugeno (15.67).

In subdirectorul MAGAYIE\fuzzy se găsesc mai multe exemple de folosire
a elementului fuzzy. Primul program, RampMic.s are două elemente:

100ramp→200fuzy

Fişierul ataşat, cu numele RampMic.dat, din lista 9.1, are o singură regulă:

Dacă intrarea este **Mic** → comanda = 1

Termenul fuzzy **Mic** este definit în universul valorilor reale de la 1 la 10 prin funcţia sa de apartenenţă μ(x) sub formă de trapez. Cele patru colţuri ale trapezului de coordonate (x, μ) sunt determinate de punctele (0, 0), (1, 1), (3, 1) şi (4,0). Fişierul din lista 9.1 conţine coperta de 20 linii, o linie cu valoarea 1 care indică numărul de reguli şi o ultimă linie a cărei prime opt valori sunt tocmai colţurile trapezului. Numărul al nouălea indică valoarea iniţială a lui μ, aici zero. Ultimul număr este comanda egală cu 1 în acest caz. Rezultatul simulării este prezentat în *Fig. 8.13*.

Lista 9.1

```
1 : Fişierul cu o singura regula pentru prog. RampMic.s
2 :

19 :
20 :
1
0 0 2 1 2 1 4  0 0 1
```

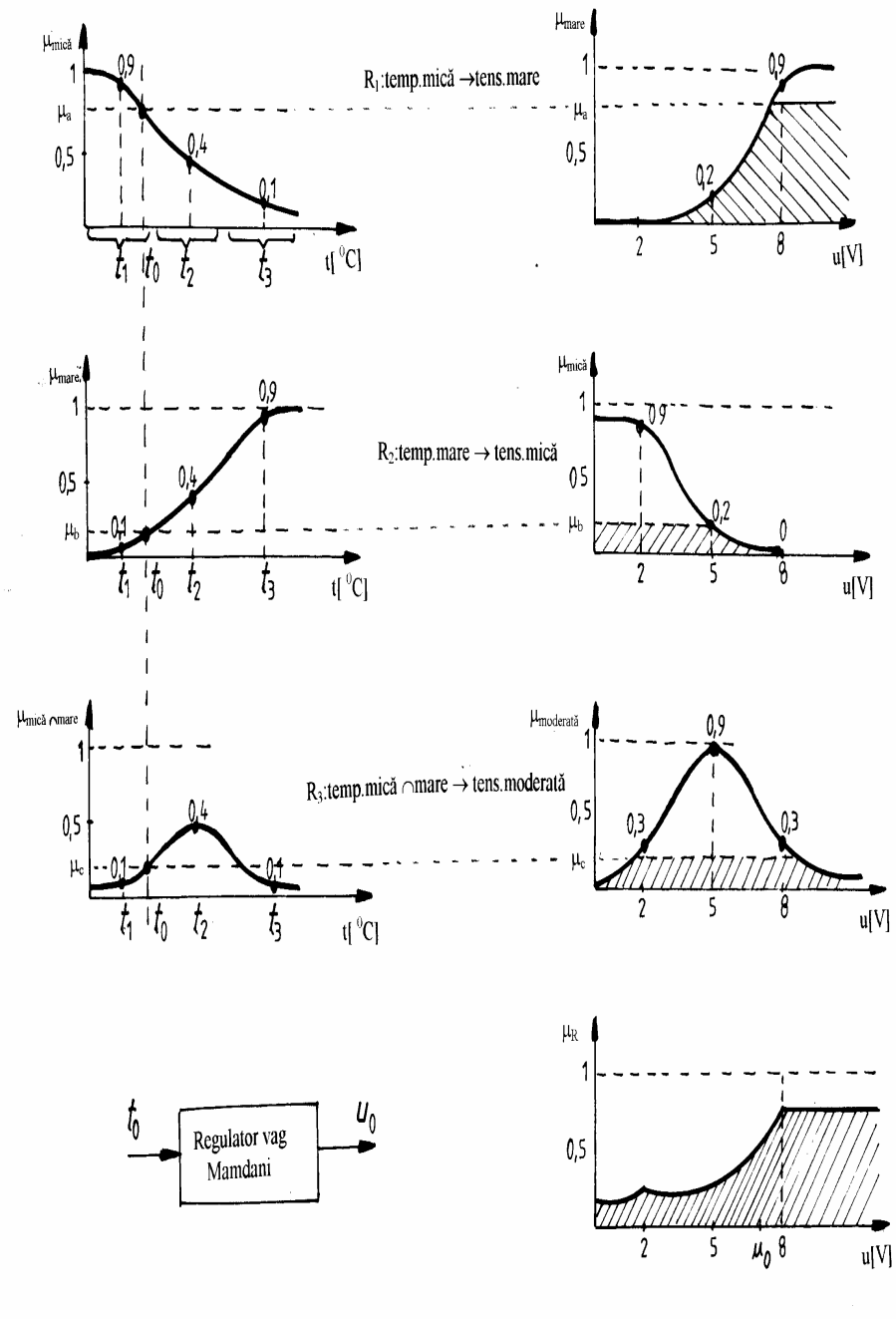


Fig. 8.11 Determinarea comenzii regulatorului fuzzy tip Mamdani

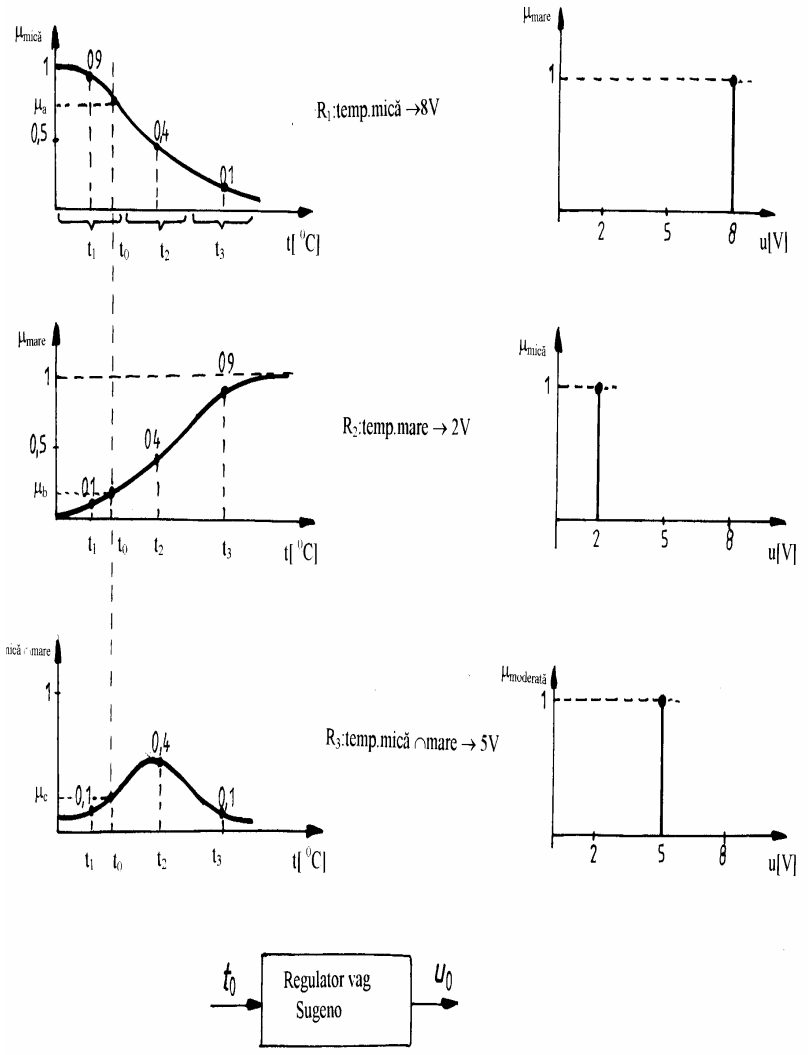


Fig. 8.12 Determinarea comenzii regulatorului fuzzy de tip Sugeno

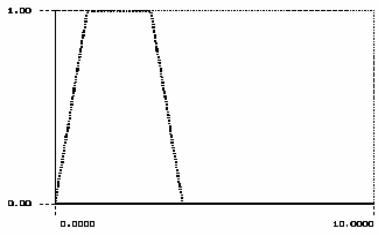


*Fig. 8.13 Funcţia de apartenenţă μ pentru termenul **Mic***

8.2. Regulatorul PID fuzzy.

Denumirea de 'Controler fuzzy' reprezintă o preluare cu modificări minime a termenului englez 'Fuzzy Controller' şi pare să sugereze în numeroase prospecte o proprietate care îmbunătăţeşte performanţele sale. Realitatea este mai nuanţată şi ar putea fi percepută mai clar dacă s-ar folosi traducerea românească: 'Sistem de conducere bazat pe logica fuzzy'. Mărimile fuzzy nu pot să fie mai folositoare decât cele nefuzzy, clare, dacă acestea există. De altfel, mărimea clară este un caz particular al unei mărimi fuzzy şi se poate demonstra că toate operaţiile în acest caz sunt mai simple şi mai precise. Dar nu avem întotdeauna la dispoziţie mărimi nefuzzy, în special atunci când omul este implicat puternic în sistemul de conducere şi atunci, bineînţeles că soluţia bazată pe logica fuzzy este singura disponibilă. Aceasta devine mai clar dacă precizăm ce înseamnă un sistem de conducere. Obiectele componente ale unui astfel de sistem sunt de diferite tipuri, cum ar fi: automat, compensator, expert, supraveghetor, identificator, etc. Implicarea omului în aceste componente este diferită, ea va fi mult mai mare într-un sistem expert, de exemplu. Astfel că este posibil ca un sistem de conducere modern să fie constituit dintr-un compensator şi un identificator clasice şi un expert bazat pe logica fuzzy care ajută la iniţializarea şi acordarea lor. Deci folosirea termenului de controler fuzzy sau a traducerii româneşti, sistem de conducere bazat pe logica fuzzy, nu permite o imagine mai conturată a dispozitivului şi compararea sa cu alte realizări. Pentru aceasta este necesară precizarea tipului de sistem de conducere la care se foloseşte logica fuzzy.

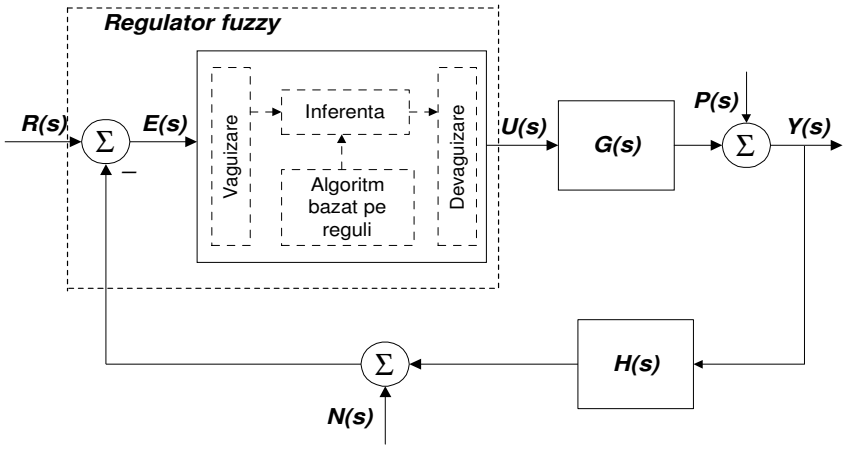


Fig. 8.14 Sistem cu regulator fuzzy

In continuare mă voi referi la compensatoarele fuzzy care sunt propuse uneori ca soluţii mai bune pentru conducerea proceselor. O schemă bloc simplificată de utilizare a unui astfel de compensator este prezentată în Fig. 8.14.Se observă că mărimile implicate în procesul de conducere, referinţa R, ieşirea Y, eroarea E, comanda U, nu sunt fuzzy. De obicei ele sunt prezente sub forma unor semnale unificate, 0..10V sau 4..20mA. Din această cauză la intrarea şi ieşirea compensatorului erorii se folosesc un dispozitiv de fuzzyficare şi unul de

defuzzyficare. Algoritmul de conducere, dacă reproduce experiența unui operator uman, poate fi bazat pe logica fuzzy. Fuzzyficarea însă a unor algoritmi clasici, cum ar fi PID, este cel puțin îndoielnică. Iată argumentele.

Compensatorul fuzzy este implementat întotdeauna numeric, cu ajutorul unui automat programabil sau microcalculator. Aceasta înseamnă că mărimea de intrare în compensator, eroarea E, este eşantionată cu perioada T. Valoarea lui T este un nou parametru al acordării regulatorului PID. O alegere necorespunzătoare a perioadei de eşantionare poate conduce la fenomene de instabilitate numerică sau la o acordare care nu asigură performanțele necesare. La fel de supărător este şi faptul că dacă se modifică eşantionarea trebuie să se modifice şi ceilalți parametrii ai compensatorului. Şi în cazul compensatorului PID fuzzy aceştia sunt mult mai mulți.

În Fig. 8.15 se prezintă toate componentele compensatorului fuzzy din Fig. 8.14. Eroarea E şi mărimea de comandă U sunt descrise prin următorii 5 termeni lingvistici: NL - negativ mare, NS - negativ mic, ZE – zero, PS – pozitiv mic şi PL – pozitiv mare. Funcțiile de apartenență a erorii E la termenii lingvistici au o formă triunghiulară şi sunt prezentate în partea superioară a Fig. 8.15. Pantele laturilor triunghiurilor pot fi modificate ca în Fig. 8.16. Funcțiile de apartenență ai mărimii de comandă U la termenii lingvistici sunt de tip 'singleton' şi apar tot în partea superioară a Fig. 8.15 şi Fig. 8.16. După cum se observă şi aceste funcții pot fi modificate.

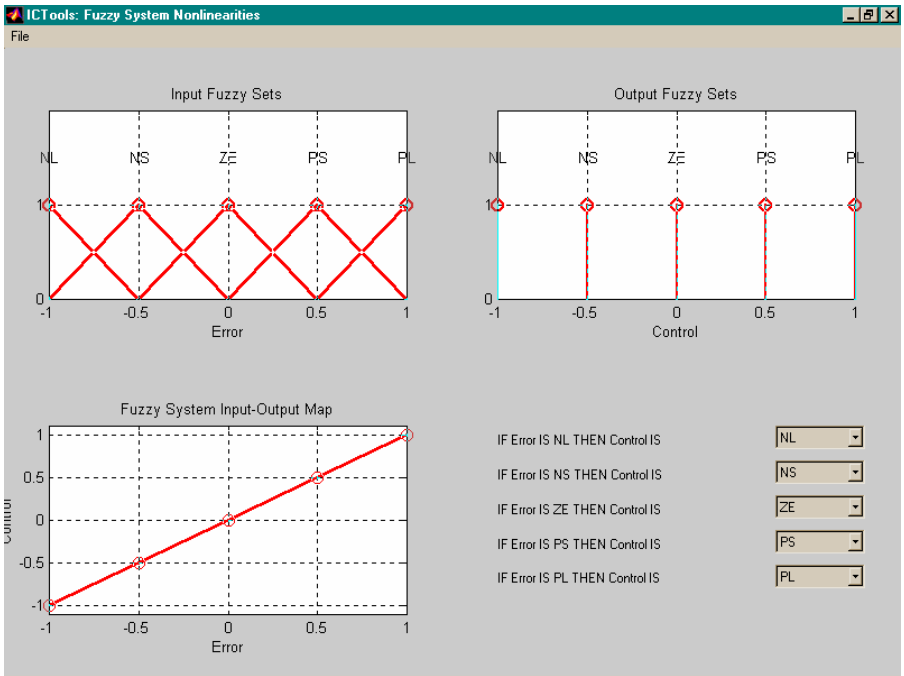


Fig. 8.15. Funcția de transfer liniară a compensatorului fuzzy.

Baza de reguli cuprinde 5 reguli prezentate în partea dreaptă jos a Fig. 8.15 şi Fig. 8.16. În termeni lingvistici prima regulă spune că dacă eroarea este negativă şi mare (NL) atunci şi mărimea de comandă trebuie să fie negativă şi mare.

Cu aceste reguli şi funcţii de apartenenţă se calculează funcţia de transfer a compensatorului fuzzy reprezentată grafic în partea stângă jos din Fig. 8.15. După cum se vede funcţia de transfer este liniară, rezultând un compensator de tip P cu constanta de proporţionalitate $K_P = 1$.

Dacă se modifică regula a patra şi funcţiile de apartenenţă pentru intrarea şi ieşirea din compensatorul fuzzy ca în Fig. 8.16 rezultă o nouă funcţie de transfer prezentată în partea din dreapta jos din Fig. 8.16. Se observă că de data aceasta funcţia de transfer este neliniară. Mai mult, pe o porţiune panta curbei este negativă. Aceasta poate provoca instabilitatea sistemului automat!

Se poate face acum o comparaţie între regulatorul fuzzy din Fig. 8.15 şi Fig. 8.16 şi un regulator proporţional clasic.

- Regulatorul tip P clasic necesită doi parametrii de acordare: constanta de proporţionalitate K_P şi perioada de eşantionare T. Regulatorul fuzzy studiat şi care este o variantă simplificată necesită 16 parametrii de acordare: perioada de eşantionare, 5 funcţii de apartenenţă pentru intrare, 5 funcţii de apartenenţă pentru ieşire şi 5 reguli.
- Regulatorul fuzzy este în general neliniar. Stabilitatea şi robusteţea sistemului automat nu poate fi analizată şi asigurată prin proiectare. Mai mult, s-a arătat că în unele cazuri neliniaritatea este de aşa natură încât este foarte probabil ca sistemul automat să fie instabil.
- Termenii lingvistici utilizaţi în stabilirea algoritmului fuzzy al compensatorului sunt în practică mult mai mulţi decât în exemplul considerat. Semnificaţia lor nu este, contrar aşteptărilor, înţeleasă uşor de către utilizator. Acelaşi lucru se poate spune şi despre reguli.
- Regulatoarele clasice pot fi acordate experimental la *in situ.* Regulatoarele cu compensatoare fuzzy necesită un program special pentru definirea funcţiilor de apartenenţă şi acordarea impune simulări repetate.
- Spre deosebire de regulatoarele clasice, pentru sistemele automate cu regulatoare fuzzy nu există metode de analiză şi proiectare analitice. Se utilizează proiectarea prin încercări şi verificări cu ajutorul simulării.
- Mulţi operatori cu pregătire medie ai instalaţiilor tehnologice au dificultăţi cu acordarea şi întreţinerea sistemelor automate cu regulatoare PID. Statisticele arată că din această cauză multe bucle de reglare automată sunt deschise prin punerea regulatorului pe poziţia de comandă manuală. cum credeţi că vor reacţiona aceşti operatori faţă de regulatoarele fuzzy care adesea au algoritmi mai complicaţi decât cel exemplificată?
- Datorită complexităţii calculelor, regulatoarele fuzzy necesită perioade de eşantionare de zece ori mai mari decât regulatoarele numerice convenţionale. Practic aceasta înseamnă că dacă nu se folosesc

procesoare specializate, mai scumpe, aplicarea lor se restrânge la conducerea proceselor lente.

- In trecut, unele regulatoare sau automate fuzzy erau mai ieftine decât echivalentul lor convențional. Din această cauză au fost folosite la unele bunuri de larg consum. In prezent se folosesc microcalculatoare în ambele situații şi acest avantaj pentru controlerele fuzzy, sau fuzzy, conform denumirii comerciale, a dispărut.

Totuşi regulatoarele fuzzy au un viitor. Nu prin înlocuirea regulatoarelor numerice convenționale. Ele pot conține nu compensatoare fuzzy, ci identificatoare sau experți bazați pe logica fuzzy. Există de asemenea situații în care nu se pot folosi algoritmi de tip convențional, de exemplu PID. Aceste cazuri se referă de obicei la procese multivariabile, neliniare, imprevizibile sau insuficient cunoscute. Chiar şi în aceste situații însă, algoritmii bazați pe logica fuzzy sunt combinați cu alți algoritmi, convenționali sau nu. Transparența modului de funcționare a unor astfel de sisteme automat pentru operatorul cu pregătire medie va fi însă foarte redusă. Proiectarea şi întreținerea lor vor necesita sisteme de programe speciale care să faciliteze interacțiunea cu utilizatorul.

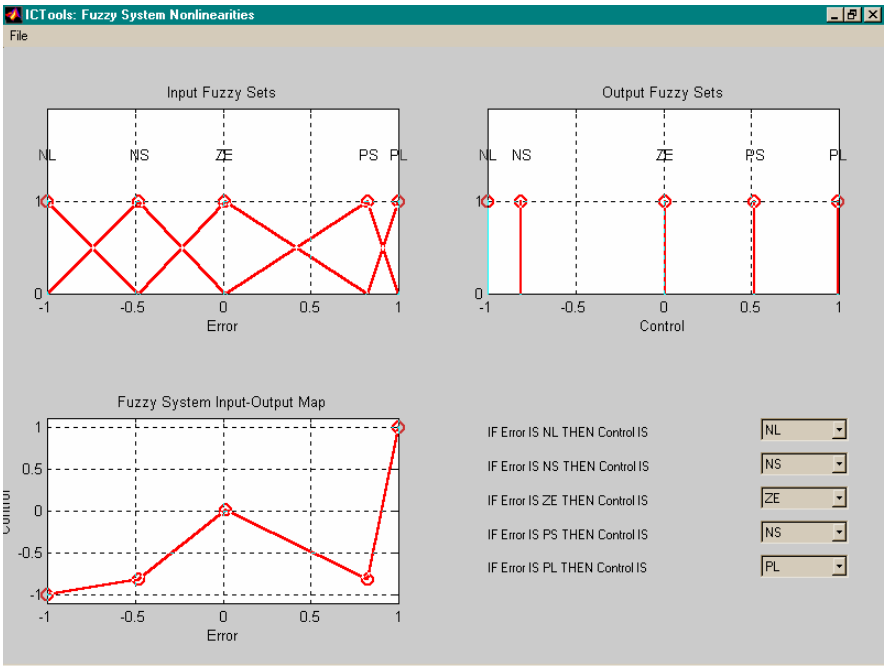


Fig. 8.16 Funcția de transfer neliniară a compensatorului fuzzy.

9. Sisteme de domotică şi imotică

Sistemele de domotică si imotica sunt sisteme definite în clădiri şi locuinţe a căror elemente îl constituie subsisteme fizice, de conducere automată şi informatice. Sistemele de domotică moderne sunt conduse de unul sau mai multe calculatoare[179] specializate[180] numite controlere, automate programabile[181] sau calculatoare de proces. În varianta cea mai simplă microcalculatorul este un automat sau regulator care realizează algoritmul de conducere. Acesta poate fi ales dintr-o gamă largă care are la bază algoritmii deja prezentaţi: automat secvenţial[182], P, PI, PD şi PID. Folosirea calculatoarelor în conducerea proceselor pune însă probleme suplimentarea. Dintre acestea cea mai dificilă este legată de tipul finit de calcul a algoritmului de conducere. Cât timp se calculează algoritmul calculatorul nu mai primeşte informaţii despre starea sistemului şi nu trimite comenzi pentru conducerea sistemului. Din această cauză semnalele de intrare şi de ieşire din calculator sunt luate în considerare periodic numai la momente discrete de timp. Această operaţie se numeşte ***eşantionare***. Dacă eşantionarea se face cu o perioadă prea mare informaţiile despre sistem sunt mai puţine şi conducerea sa este mai proastă.

Sistemele de domotică si imotica sunt sisteme de conducere automată a echipamentelor, instalaţiilor şi serviciilor din locuinţe şi imobile[183]. Aceste sisteme sunt caracterizate printr-o distribuţie în spaţiu şi o structură ierarhică. Frecvent, în ţara noastră şi în ţările de limbă latină acestea sunt numite doar sisteme de domotică, deşi conducerea automată pentru clădirile mari (imobile de tipul spitalelor, hotelurilor şcolilor, etc.) este mai importantă şi mai dificilă decât conducerea automată pentru locuinţe.

Distribuţia spaţială a sistemelor de domotică şi imotică implică transmiterea informaţiilor prin fir, magistrală (bus) sau prin radio. În prezent sunt frecvente sistemele de domotică şi imotică care folosesc comunicaţia prin reţele informatice cu suport electric format dintr-un bus (magistrală).

Caracterul ierarhic al sistemelor de domotică şi imotică este evidenţiat de protocoalele de comunicare. Pentru sistemele de domotică şi imotică cele mai importante protocoale sunt CEN TC247 şi BACnet care prevăd trei nivele ierarhice:

- management,

---

[179] Sistemele de conducere mai simple folosesc microcalculatoare.

[180] Spre deosebire de calculatoarele personale PC acestea sunt destinate să lucreze în medii neprietenoase, cu multe perturbaţii, iar condiţiile de fiabilitate sunt mult mai severe. Deşi sunt realizate constructiv în mod diferit principiile de funcţionare sunt în mare aceleaşi.

[181] Automatul programabil logic APL este un automat programabil AP particular care lucrează numai cu semnale binare. Automatul programabil conţine de multe ori două microcalculatoare specializate, unul calculează algoritmul şi altul realizează protocolul de comunicaţie.

[182] Automatul secvenţial prezentat a fost de tip bipoziţional 2P. Alte automate secvenţial sunt folosite foarte frecvent, cum ar fi: automatul pentru pornirea directă a unui motor electric, automatul pentru pornirea stea – triunghi a unui motor, etc.

[183] Systeme domotique et immotique, Home and building automation.

- automatizare,
- câmp (procese).

Dacă sistemul este condus de mai multe calculatoare este necesar să se stabilească o ierarhie a acestora. Dacă aceste calculatoare sunt distribuite în spaţiu la distanţe semnificative trebuie să se rezolve problema comunicaţiei dintre ele. Aceasta se face printr-o structură cu trei nivele ierarhice ca în Fig. 9.1.

Nivelul superior de conducere este de tip supraveghere şi gestiune tehnică (management) şi poate fi format dintr-o reţea locală LAN de calculatoare şi alte echipamente. Schemele sinoptice apar pe calculatoarele de la acest nivel. Tot aici sunt concentrate şi facilităţile de obţinere a unor rapoarte tipărite şi de comunicare la mare distanţă prin internet, radio sau telefon. De obicei supravegherea se face dintr-o cameră specială situată la o distanţă de instalaţiile distribuite în clădire. Un sistem real de management (gestiune tehnică) necesită o bază de date ca în Fig. 9.1.

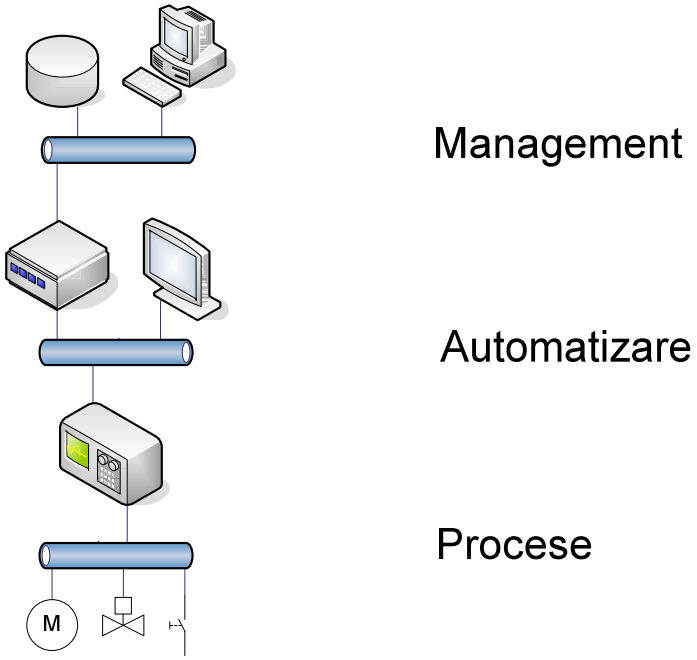


Fig. 9.1. Structura unui sistem automat ierarhic şi distribuit.

Nivelul al doilea al sistemului ierarhic este ocupat de calculatoarele care realizează funcţiile de conducere automată. De cele mai multe ori aceste calculatoare sunt de tipul automate programabile AP sau automate programabile specializate pentru securitate, control acces şi supraveghere şi alarmare la incendiu. Unele dintre aceste automate programabile pot fi conectate într-o reţea locală sau pot fi legate direct la calculatorul de supraveghere.

Al treilea nivel ierarhic este ocupat în Fig. 9.1 de traductoare şi elemente de execuţie. Acestea pot fi într-un număr foarte mare într-o clădire şi comunicarea

dintre ele şi automatele programabile pun probleme legate atât de cablaj cât şi de protocol de comunicaţie. O soluţie modernă constă în folosirea unor aparate inteligente care conţin fiecare câte un microcalculator.

Toate echipamente situat pe un anumit nivel ierarhic sunt interconectate cu ajutorul unei magistrale (bus).

Folosirea calculatoarelor pentru conducerea automată a sistemelor este avantajoasă nu atât pentru calculul algoritmului, care am văzut că poate produce dificultăţi datorită eşantionării, cât pentru noile funcţiuni pe care le pot îndeplini. Printre aceste menţionăm: 1) Achiziţia, prelucrarea şi memorarea unui volum imens de date, 2) Accesul rapid la datele memorate, 3) Realizarea unei interfeţe grafice interactive şi prietenoase cu utilizatorul, 4) Instruirea operatorului şi ajutarea sa printr-un sistem expert la apariţia unor defecţiuni, 5) Autotestarea sistemului automat şi autoacordarea regulatorului, 6) Autoacordarea automată adaptivă a regulatorului, 7) Combinarea mai multor sisteme conduse cu calculator într-o structură ierarhică, 8)Comunicarea la distanţă cu alte calculatoare, 9) Transformarea traductoarelor şi elementelor de execuţie în aparate inteligente. Aceste funcţiuni noi apar adeseori împreună şi creează aparate şi sisteme de conducere cu denumiri specifice.

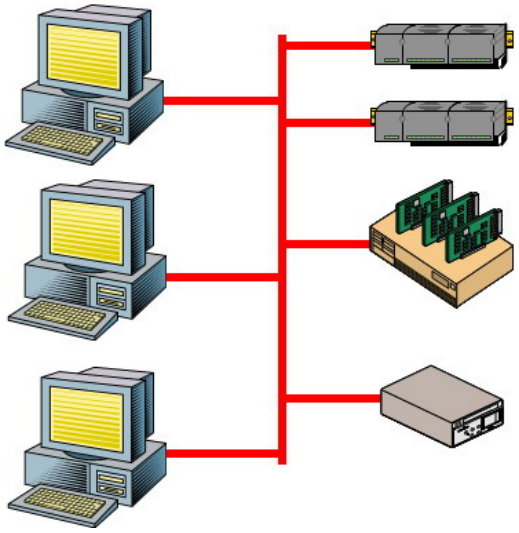


Fig. 9.2 Fiecare PC accesează fiecare dintre dispozitivele intrare/ieşire

Modul de conectare al calculatoarelor di dispozitivelor de intrare/ieşire la o reţea este prezentat în Fig. 9.2, Fig. 9.3 şi Fig. 9.4. În prezent echipamentele au calculatoare specializate care controlează separat accesul la reţea şi accesul la dispozitivele de intrare/ieşire.

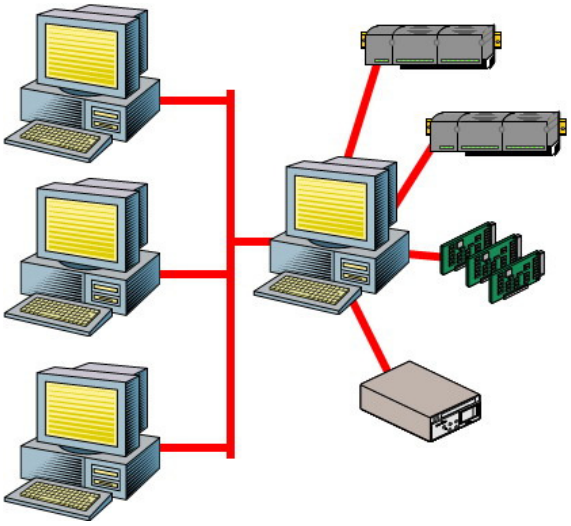


Fig. 9.3 Un singur calculator accesează fiecare dintre dispozitivele de intrare/ieşire

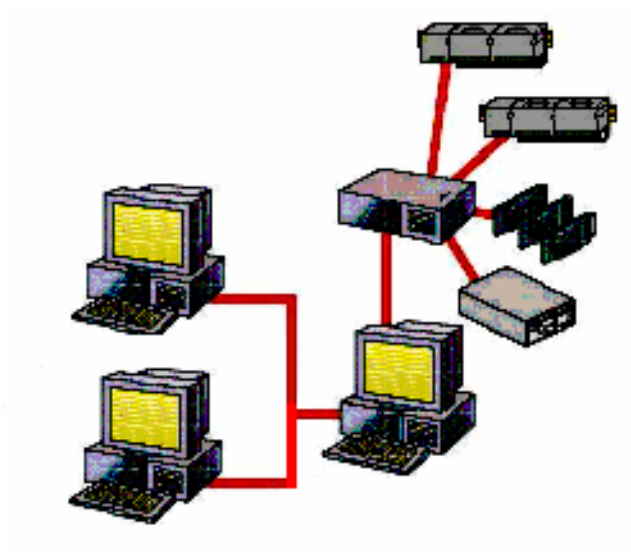


Fig. 9.4 Un calculator controlează intrările şi ieşirile iar altul reţeaua

9.1. Sisteme informatice

Multe sisteme de domotică au conectat nivelul de management la un sistem informatic.

Elementele unui sistem informatic sunt datele[184], procesele[185], echipamentele informatice şi fluxurile de date[186]. Scopul unui sistem informatic îl constituie achiziția, prelucrarea datelor şi prezentarea datelor. Sistemele informatice evită supraîncărcarea sau gâtuirea informativă, ajută la luarea deciziilor şi organizarea activităților.

Sistemele informatice sunt definite, în prezent, în patru feluri:

- Sisteme informatice orientate spre date şi construite în jurul unor bazele de date relaționale.
- Sisteme informatice orientate spre procese şi construite din elemente tip intrare – transformare – ieşire.
- Sisteme informatice orientate spre obiecte[187] care conțin atât structuri de date cât şi procese.
- Sisteme informatice orientate spre fluxuri. De exemplu rețelele de calculatoare. Relațiile dintre elementele sistemului (calculatoarele) pot fi definite la nivel hardware sau software şi formează diferite structuri: stea, magistrală, arbore, etc. Scopul rețelelor de calculatoare îl constituie comunicarea dintre calculatoare. Internetul şi intranetul sunt sisteme informatice.

Exemple tipice de sisteme informatice se pot definii într-o organizație. În general activitatea dintr-o organizație este structurată pe cinci direcții:

- Servicii şi fabricare.
- Vânzări şi marketing.
- Financiar.
- Contabilitate.
- Resurse umane.

Pentru fiecare direcție se pot definii diferite sisteme informatice. De exemplu, pentru activitatea de servicii şi fabricare se folosesc, la diferite nivele ierarhice, următoarele sisteme informatice:

9.1.1. Sisteme informatice la nivelul de planificare

Sistem de suport al planificării[188]. Cu ajutorul lui se planifică operațiile de servicii şi fabricare pe termen lung, de exemplu cinci ani. Intrările sunt formate di volume mari de date furnizate de magazie. Ieşirile sunt diferite analize şi proiecții folositoare directorilor.

---

[184] Datele sunt informații prelucrate. Legătura dintre date şi informații este studiată de către semiotică. În cadrul sistemelor informaționale datele sunt organizate în fişiere.

[185] Procesele de transformare a datelor şi evenimentele care le declanşează.

[186] Fluxurile de date sunt grupuri de date care se deplasează în sisteme şi care includ descrierea sursei şi destinației fluxului.

[187] Obiectul este o structură care încapsulează (împachetează) atribute (date specifice) şi metode care operează cu aceste atribute.

[188] ESS – Executive Support System.

9.1.2. Sisteme informatice la nivelul de management

Sistem informatic de management[189]. Acesta realizează controlul materiilor prime, energiei, inventarului şi evenimentelor semnificative[190] apărute în timpul serviciului. Un exemplu îl constituie sistemul de management al clădirilor[191]. Intrările sunt formate din volume mari de date şi din această cauză sunt folosite baze de date. prelucrările se bazează pe modele simple. ieşirile sunt formate din rapoarte simple.

Sistem de suport al deciziilor[192]. Orarul de activitate este realizat cu ajutorul acestui sistem. Intrările sistemului de suport al deciziilor provin din diferite baze de date din organizaţie şi din afara organizaţiei[193]. Prelucrările sunt interactive folosind diferite interfeţe utilizator[194] şi folosesc modele sofisticate, inclusiv matematice. Aceste sisteme nu sunt automatizate, trebuie să fie flexibile, adaptabile, rapide şi folosesc la luarea deciziilor răspunzând la întrebări de tipul: ce se întâmplă dacă?

9.1.3. Sisteme informatice la nivelul de cunoaştere.

Sisteme expert. Cu ajutorul acestor sisteme se poate acorda asistenţă tehnică, de exemplu, la efectuarea reparaţiilor.

Sisteme birotice. Sistemele de elaborare şi prelucrare a textelor, desenelor şi calendarele electronice fac parte din această categorie.

Sisteme de simulare. Intrările provin din date de proiectare, identificări experimentale şi măsurări on şi off line. Prelucrarea este de tipul simulare iar ieşirile sunt de tipul desene, grafice, rapoarte, documente, etc.

9.1.4. Sisteme informatice la nivelul operaţional

Sisteme de prelucrare a tranzacţiilor. Pentru departamentul de producţie şi servicii intră în această categorie planul de reparaţii al instalaţiilor şi utilajelor, sistemul de achiziţie a datelor şi conducere de supraveghere[195], sisteme de control a calităţii, etc. Intrările în aceste sisteme sunt formate din diferite tranzacţii[196] şi evenimente semnificative. Prelucrarea intrărilor este de obicei simplă, de tipul actualizarea bazei de date. Ieşirile sistemului îl constituie rapoartele detaliate destinate personalului operativ.

Sistemele informatice din departamentul tehnic al clădirilor sunt, de obicei, integrate în sisteme de domotică. Pe de altă parte aceste sisteme trebuie să fie în legătură şi cu sistemele informatice din celelalte departamente care asigură funcţionarea organizaţiei.

---

[189] MIS – Management Information System.

[190] Astfel de evenimente sunt avariile, tentativele de furt, situaţii periculoase care pot provoca incendii, etc.

[191] BMS – Building Management System.

[192] DSS – Decision Support System.

[193] De exemplu evaluarea riscului de ţară.

[194] GUI – Graphical User Interface.

[195] SCADA – Supervisory Control And Data Acquisition System.

[196] Transformări, operaţii.

9.2. Sisteme de domotică cu structuri particulare

Sisteme de domotică cu structuri particulare formează cunoscutele sisteme de conducere automată, dintre care prezentăm câteva exemple.

9.2.1. Sistem de domotică pentru conducerea automată

Sistemul de reglare automată este format din unul s-au mai multe regulatoare legate în cascadă împreună cu traductoarele şi elementele de execuţie corespunzătoare. Regulatorul, elementul de execuţie şi traductorul formează o buclă. Într-o clădire există în mod frecvent mai multe bucle de reglare automată. Toate aceste bucle, luate împreună, nu formează un sistem de domotică deoarece nu comunică între ele prin intermediul uni bus.

Sistemul de conducere cu evenimente discrete este format de asemenea dintr-un automat programabil logic – APL împreună cu traductoarele şi elementele de execuţie aferente. Mai multe APL într-o clădire nu formează un sistem de domotică fiindcă nu sunt interconectate prin intermediul unei magistrale (bus)

Dacă în clădire există cel puţin două echipamente de tip regulator sau APL care comunică între ele printr-un bus avem un sistem de domotică pentru conducerea automată. În acest caz în sistemul de domotică există numai nivelul ierarhic de automatizare, unde se găsesc regulatoarele şi automatele şi nivelul ierarhic de câmp unde se găsesc traductoarele şi elementele de execuţie. Şi la nivelul de câmp comunicarea se poate face prin intermediul unei magistrale. Regulatoarele şi automatele programabile logice pot fi realizate fizic cu ajutorul unor automate programabile – AP sau cu microcalculatoare de proces.

În sistemul de domotică pentru conducerea automată poate un exista şi un calculator situat la nivelul ierarhic superior, de management, care sa stabilească referinţele regulatoarelor. Acest sistem de domotică nu este însă un BMS – Building Management System deoarece calculatorul nu realizează operaţii specifice de management. Aceste operaţii necesită, în general, prezenţa unei baze de date.

9.2.2. Sistem de domotică pentru gestiunea clădirii - BMS

Unul sau mai multe calculatoare care gestionează orarul în campusul unei universităţi formează un sistem de gestionare a serviciilor oferite de clădiri fără să fie un sistem de domotică. Acest sistem se bazează, în esenţă, pe o bază de date şi pe comunicarea în cadrul intranet şi internet. În acest caz în sistem există numai nivelul de management dar nu exista nivelul de automatizare şi de câmp. Dacă calculatorul stabileşte şi referinţele regulatoarelor de temperatură în funcţie de orar se obţine un sistem de domotică. Asemănător, sistemul de gestiune pentru orar poate fi integrat cu un sistem de educaţie şi în acest caz se obţine un sistem de gestionare a cunoştinţelor CMS – Content Manager System. Remarcăm că un BMS trebuie să aibă o bază de date.

9.2.3. Conducerea tip SCADA/HMI[197]

Un sistem care are doar nivelul de management şi nivelul de câmp poate forma un sistem SCADA. Acesta achiziţionează date şi le prelucrează, eventual cu ajutorul unei baze de date, dar prelucrarea nu se face în timp real. Posibilităţile grafice deosebite şi capacitatea de funcţionare în mod interactiv a calculatoarelor moderne permit realizarea unor interfeţe care să faciliteze la maximum activitatea operatorului. Punctul central al interfeţei îl constituie schema sinoptică[198]. Obiectele din schema sinoptică permit, atunci când sunt selectate, afişarea unor noi scheme sinoptice sau a unor grafice şi tabele care prezintă în mod interactiv[199] starea sistemului pe baza informaţiilor acumulate prin achiziţia şi memorarea datelor furnizate de traductori. Chiar dacă operatorul nu întreprinde nici o acţiune, atunci când aceste informaţii ies dintr-un anumit domeniu se produc semnalizări şi alarme. În funcţie de informaţiile obţinute operatorul poate realiza manual diferite operaţii de conducere. De exemplu poate porni sau opri anumite pompe cu ajutorul întrerupătoarelor prezentate alăturat în schema sinoptică.

9.2.4. Sistem de domotică pentru securitatea clădirii.

Sistemul de securitate este format, de obicei, dintr-un automat programabil logic împreună cu traductoarele şi elementele de execuţie necesare. Dacă sistemul de securitate comunică (eventual la distanţă) cu un calculator care are o bază de date se obţine un sistem de domotică simplu. Calculatorul situat la nivelul de management poate să comunice cu alte automate, de exemplu cu cele care comandă ascensoarele şi să comande acţiuni specifice în situaţii de alarmare speciale.

9.2.5. Sistem de domotică pentru siguranţa clădirii

Componentele unui sistem de siguranţă la incendiu şi funcţiunile îndeplinite sunt prezentate în figura următoare. Centrala sistemului de siguranţă este situată la nivelul de automatizare iar echipamentele de intrare şi ieşire sunt situate la nivelul de câmp. În cele mai multe cazuri centrala este conectată la un sistem de gestiune a evenimentelor aflat la distanţă şi pe un nivel ierarhic superior.

9.2.6. Sistem de domotică pentru gestiunea energiei

Acest sistem de domotică este în esenţă un sistem SCADA cu două nivele ierarhice, de câmp şi de management. Calculatorul situat la nivelul de management foloseşte o bază de date pentru prelucrări şi prezentări referitoare la aspectele energetice ale clădirii.

---

[197] SCADA – Supervisory Control And Data Acquisition / Human – Machine Interface.

[198] O schemă care permite observarea dintr-o privire a tuturor componentelor sistemului.

[199] Funcţionarea interactivă calculator – operator se realizează practic de cele mai multe ori în modul următor: operatorul selectează un obiect sau o acţiune prezentate în schema sinoptică iar calculatorul îi prezintă o serie de instrucţiuni şi operaţiuni posibile în continuare.

9.3. Protocoalele sistemului de domotică

Regulile de integrare a echipamentelor în clădiri şi locuinţe, adică protocoalele, oferă următoarele avantaje:

- Echipamente produse de fabricanţi diferiţi pot folosi acelaşi mediu de comunicare (cabluri, radio, etc)
- Pun în evidenţă structura ierarhică a sistemului,
- Diferitele servicii pe care le oferă clădirea şi echipamentele sale (încălzire, ventilare, iluminare, securitate la efracţie, siguranţă la foc, etc) pot fi integrate.
- Toate serviciile folosesc o singură interfaţă pentru operator
- Independenţă de furnizorul iniţial de echipamente în cazul unor extinderi ale sistemului.

9.3.1. Protocoalele CEN TC247 şi BACnet

Există foarte multe protocoale, de firmă, internaţionale, guvernamentale, asociaţii internaţionale de standardizare. În România au importanţă recomandările Standardului Comunităţii Europene, Comitetului Tehnic 247 (CEN TC247).

Pentru cele trei nivele ierarhice ale sistemelor de domotică standardul CEN TC247 a recomandat protocoalele din tabelul următor.

Tab. 9.1

Nivelul	***Standardul CEN TC 247***	***Data***	***Proto-colul***	***Mediul de transmitere***
Managemen t	ENV 1805-2	1998	BACnet	Ethernet – toate tipurile de media Modem Dial up
Automa-tizare	ENV 13321-1	1999	PROFIBUS	Pereche răsucită
			BACnet	Ethernet – toate tipurile de media Modem Dial up
	ENV 13321-2	2000	EIB (Instabus)	Ethernet – toate tipurile de media
Câmp	ENV 13154-2	1998	EIB (Instabus)	Pereche răsucită Reţea alimentare
			EHS	Pereche răsucită Coaxial Radio
			LONtalk	Pereche răsucită Reţea alimentare Radio

Câteva standarde de protocoale sunt folosite în mod curent pentru clădiri şi locuinţe fără a fi recunoscute de CEN TC 247. Printre acestea se numără MODBUS şi standardele internetului cum ar fi HTTP şi TCP-IP. Acesta din urmă a inclus recent în BACnet.

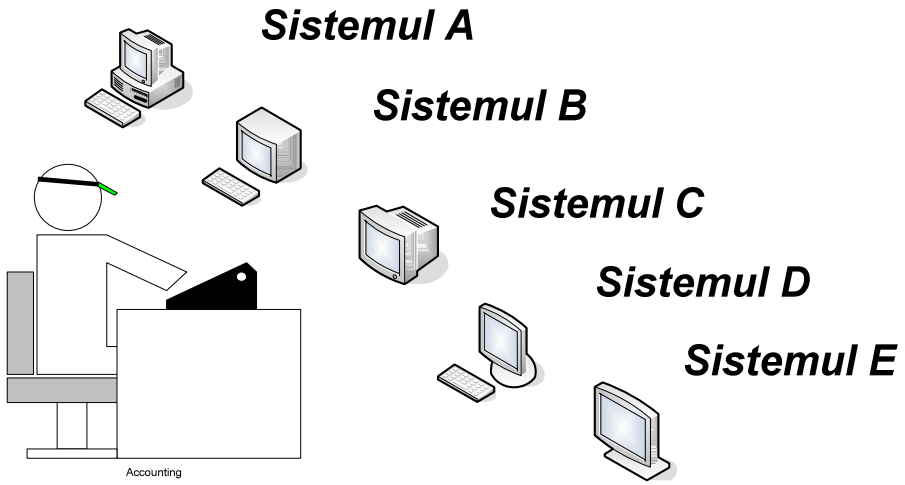


Fig. 9.5 Operatorul unor sisteme neintegrate

BACnet (Building Automation and Control Networks) este un protocol de comunicaţie elaborat de Comitetul de Proiecte de Standarde 135P al ASHRAE (American Society of Heating, Refrigrating and Air – conditioning Engineers) în anul 1994. De atunci au apărut mai multe versiuni noi, una importantă în 1997. Acest protocol furnizează mecanismele prin care orice echipament inteligent, adică un echipament ce include şi un microcalculator, poate schimba informaţii cu alte echipamente, indiferent de funcţiile sale în cadrul clădirii.

Asigurarea independenţei faţă de diversitatea funcţionalităţilor echipamentelor şi a realizărilor constructive (Fig. 9.5) a fost obţinută prin integrarea sub protocolul BACnet (Fig. 9.6). Modelarea fiecărei instalaţii inteligente de tip BACnet se face sub forma unei colecţii de structuri de date numite *obiecte.* Fiecare obiect este caracterizat de un grup de atribute sau proprietăţi. Standardul BACnet defineşte 18 tipuri de obiecte printre care: intrări şi ieşiri analogice sau logice, referinţe, bucle, alarme, program de funcţionare.

Standardul BACnet asigură 30 de servicii pentru conducerea clădirii în cinci domenii: alarme şi evenimente, accesul la fişiere, accesul la obiecte, gestiunea la distanţă a echipamentelor şi terminalelor virtuale.

Arhitectura protocolului BACnet se bazează pe Modelul de Referinţă de Bază de Interconectare pentru Sisteme Deschise (Open System Interconnection OSI) specificat în standardul internaţional ISO 7498. Din cele 7 nivele OSI BACnet a reţinut numai patru: nivelul fizic, nivelul legătură de date, nivelul reţea şi nivelul aplicaţie. La nivelele 1 şi 2, fizic şi legătură de date, BACnet oferă mai multe variante. O posibilitate este tehnologia IEEE 802.3 / Ethernet de acces

multiplu la rețea cu sesizare de purtătoare şi detectarea coliziunilor. Altă variantă este de tip MS / TP (maşter-slave / Token – Passing) cu un nivel fizic tip EIA-485. O altă variantă se referă la legături punct-la-punct cu un nivel fizic tip EIA-232. In esență nivelele fizic şi date ale protocolului furnizează detalii despre modul în care funcționează rețeaua BACnet. Nivelul rețea specifică mecanismul de interconectare a mai multor sisteme BACnet iar nivelul aplicație descrie serviciile protocolului şi regulile de folosire a acestora.

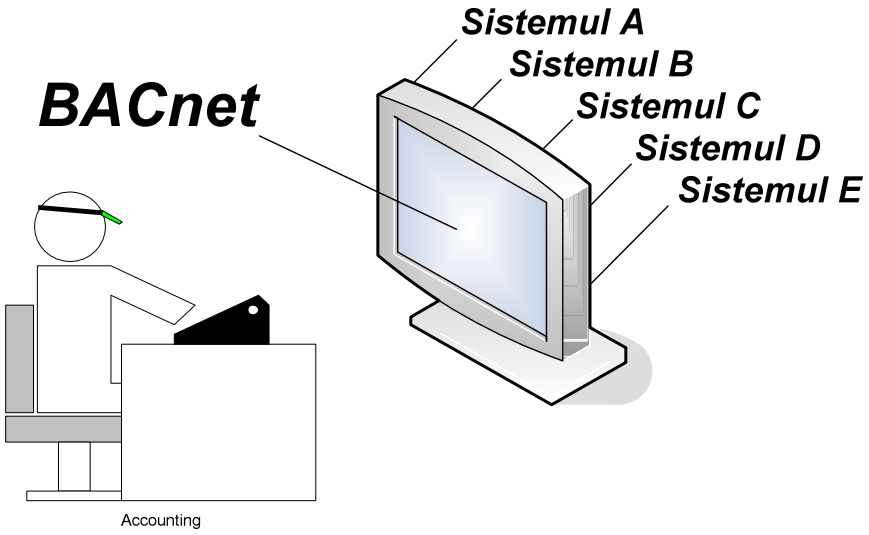


Fig. 9.6 Operatorul unor sisteme integrate sub protocolul BACnet

Standardul BACnet referitor la comunicarea dintre echipamentele unei clădiri poate fi înțeles mai bine dacă se analizează exemplul *clasic* de comunicare dintre doi filozofi prezentat în Fig. 9.7. Filozoful A vorbeşte engleza şi urdu iar filozoful B vorbeşte franceza şi chineza. Pentru ca filozofii să comunice au nevoie fiecare de un translator şi un secretar. Putem vorbi de comunicarea dintre sistemul A şi sistemul B prin mediul fizic de comunicație. În loc de două sisteme, în cazul clădirilor se pune problema comunicării între două echipamente. In ambele cazuri există mai multe nivele de comunicație. Pentru fiecare nivel se stabilesc reguli (protocoale) de comunicare, independente de celelalte nivele. De exemplu, pentru nivelul 2 translatorii au hotărât ca să existe două limbi în care să se poată face comunicarea: olandeza şi rusa. La nivelul 1 secretarii stabilesc singuri că există două sisteme fizice de comunicare: faxul şi telefonul. Se observă că emailul şi poşta sunt excluse. Deoarece mediul fizic de comunicație este linia telefonică şi este folosit de ambele sisteme, fax şi telefon, trebuie să existe o informație care să permită distincția dintre ele.

Să presupunem că filozoful A vrea să-i transmită filozofului B mesajul M prin care îl informează ca-i plac iepurii, Fig. 9.7. Pentru aceasta formulează mesajul în engleză şi-l transmite la nivelul 2 translatorului A. Acesta îl traduce în

olandeză şi adaugă la mesajul M informaţia L care specifică limba de comunicare folosită: în acest caz olandeza. Mesajul tradus ajunge la secretarul A. Acesta decide să folosească faxul şi mai adaugă informaţia E referitoare la echipament care specifică acest lucru. În final ***data*** transmisă prin mediul de comunicaţie este formată din informaţiile M,L şi E. Structura datei este stabilită de protocoalele de nivel care formează împreună protocolul de comunicaţie.

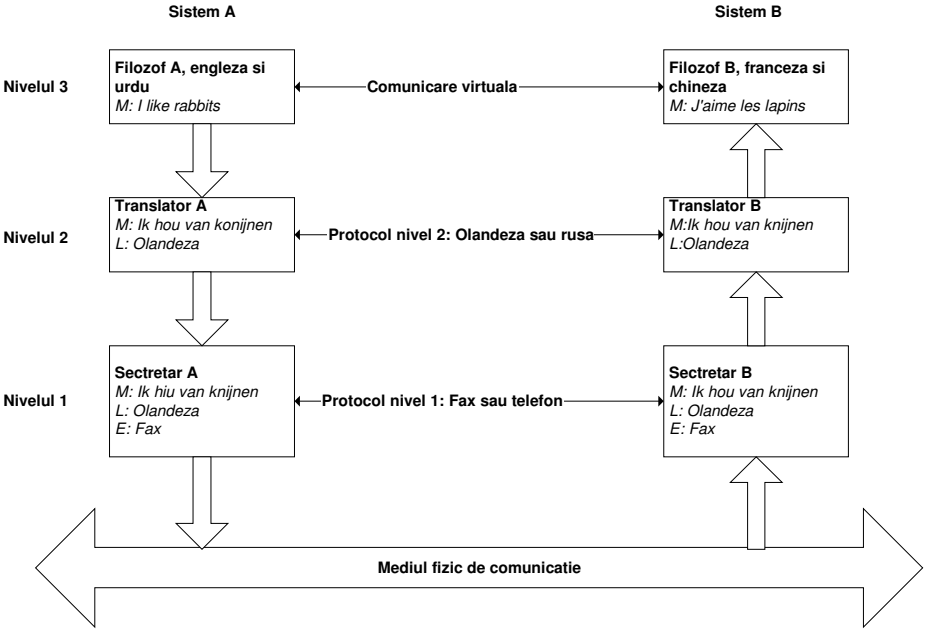


Fig. 9.7 Un protocol de comunicaţie între doi filozofi.

Pentru descrierea protocoalelor se utilizează terminologia descrisă în ISO 7498 pe baza Modelului de Referinţă de Bază de Interconectare pentru Sisteme Deschise (Open System Interconection Basic Reference Model – OSIBRM sau pe scurt OSI). Acest model pe baza căruia se descriu protocoalele, are şi el o structură ierarhizată pe şapte nivele independente între ele.

9.3.2. Protocoalele RS-232, RS-422, RS-485 şi HART

La nivelul fizic al protocoalelor se stabilesc caracteristicile mecanice, electrice funcţionale şi procedurale de acces la mediul fizic. Se consideră transmiterea de şiruri nestructurate de biţi prin mediul fizic. Exemple de standarde pentru protocolul fizic sunt RS-232C, RS-422, RS-485 şi HART.

Legătura calculatorului sau automatului programabil cu clădirea condusă se face prin intermediul următoarelor tipuri de interfeţe intrare / ieşire: plăci şi module specializate în achiziţia datelor, interfeţe paralele şi interfeţe seriale. Pentru conducerea ierarhică şi distribuită a clădirilor cele mai importante sunt interfeţele seriale care respectă standardele ANSI / EIA următoare: RS-232, RS-422 şi RS-485. Aceste standarde au următoarele caracteristici:

Tab 9.1

	RS - 232	**RS - 422**	**RS - 485**
Tipul liniei de transmisie	Nesimetrică	Diferențială	Diferențială
Lungimea maximă a cablului (m)	15,2	1200	1200
Număr de emițători	1	1	32
Număr de receptori	1	10	32
Viteza de transmisie maximă	10 kb/s	10Mb/s	10 Mb/s

Interfața RS - 232 este standardul pentru calculatoarele compatibile IBM PC. Ea este limitată la conexiunea punct - la - punct pentru distanțe mici.

Interfața RS - 422 este standardul pentru calculatoarele Aplle Macintosh. Semnalul electric este diferențial şi simetric față de pământ. Se folosesc patru fire, câte două pentru transmitere şi recepție (full-duplex). Modul diferențial de transmitere a semnalelor electrice conduce la o mai bună protecție la zgomote şi deci permite distanțe mai mari de comunicație. Aceste caracteristici sunt importante în mediul industrial cu perturbații puternice.

Interfața RS - 485 este o perfecționare a interfeței RS - 422. Acest standard permite 32 de emițători şi receptori şi defineşte caracteristicile electrice asigurării tensiunii necesare pentru sarcina maximă. Din această cauză pe baza acestui standard se pot construi rețele de aparate MPI (Multi Point) conectate la un singur port serial RS - 485. Se pot realiza comunicații full-duplex şi half-duplex. Transmisia şi recepția au loc pe aceiaşi linie de două fire, în general torsadate. Dacă unul din fire se află la potențialul +V în raport cu masa emițătorului, atunci celălalt fir va avea potențialul +V. Folosind repetori se poate creşte numărul de aparate şi distanța. Datorită imunității la zgomot şi capacității de lucru multipunct interfața RS - 485 este în prezent alegerea preferată pentru echipamente care funcționează în mediul industrial. Multe aplicații care folosesc RS-485 implementează comunicații semiduplex, în care conexiunea serială foloseşte linii comune pentru transmisie şi recepție. În achiziția de date din clădiri şi locuințe unde sunt necesare linii lungi, conexiunea semiduplex necesită mai puține fire. Bine înțeles transmisia şi recepția nu se pot face simultan. Dispozitivele din rețea sunt utilizate în configurație master / slave, unde dispozitivele slave transmit numai atunci când li se permite aceasta de către dispozitivul master (un PC sau un automat programabil). Protocolul RS-485 este un mod economic de a implementa o rețea peer-to-peer (egal-la-egal). Orice echipament din rețea trebuie să aibă inteligența pentru a recunoaşte adresa. Pot exista până la 32 de aparate pe o pereche de fire răsucite. Totuşi rețeaua este lentă deoarece numai un aparat trimite date la un moment dat.

Deoarece calculatoarele tip PC sunt echipate cu interfața RS - 232 se pot obține performanțele specificate de standardul RS - 485 în două moduri:

Se foloseşte un convertor extern RS-232 / RS-485.

Se adaugă calculatorului o placă cu interfaţa RS-485. Fiecare placă permite instalarea a patru porturi COM şi creşte dimensiunea tamponului de date. Acestea sunt avantaje faţă de soluţia precedentă a convertorului.

Standardele RS - xxx prevăd doar caracteristicile comunicaţiei la nivel fizic. Pentru specificarea tipului de software necesar transmiterii datelor sunt standardizate protocoale de comunicaţie. De exemplu, firma Simens prevede pentru aparatele sale următoarele protocol:

Tab 9.2

Nume le protocolului	***Viteza de reacţie***	***Pachetul de date***
Multi - point MPI	200 + 500 ms	76 byte
Profibus	50 + 200 ms	240 byte
Ethernet	20 + 100 ms	240 byte + 64 kbyte

Un alt standard de protocol la nivel fizic pentru achiziţia datelor este HART (Highwaz Addressable Remonte Transducer). Acesta prevede o linie de comunicaţie hibridă care foloseşte semnalul analogic 4-20 mA la care se adaugă un semnal digital pentru adresă şi comanzi, de exemplu de scriere sau de citire.

Cele mai răspândite protocoale folosesc standardul RS - 485 la nivel fizic, un grup de caractere ASCII structurat la nivelul date şi sunt implementate în mod asincron. De obicei sunt referite drept protocoale RS - 485. Reţeaua este de obicei de tip master / slave, unde există un singur stăpân, de obicei calculatorul sau automatul programabil, iar celelalte aparate joacă rolul de sclav.

In mod normal toate aparatele pornesc în modul recepţie aşteptând să primească mesaje. Atunci când stăpânul transmite un mesaj pe reţeaua multipunct toate aparatele primesc mesajul şi determină dacă îi este adresat. In acest caz se scrie pachetul de date în interfaţa serială. In mod asemănător stăpânul primeşte informaţii de la aparatele din reţea. Modul de împachetare a octeţilor într-un mesaj depinde de producătorul aparatelor tip sclav. Programul calculatorului sau automatului programabil trebuie întocmit de utilizator astfel încât să prelucreze în mod corespunzător aceste mesaje.

9.3.3. Protocoale la nivelul linie - legătura de date

Protocoalele de la acest nivel se ocupă cu transferul fără erori a datelor prin legătura fizică. Se transmit blocuri de date (cadre) cu sincronizarea necesară, controlul erorilor şi controlul fluxului. Standardul IEEE 802 consideră că nivelul linie este format din două subniveluri: controlul accesului la mediu (MAC – Medium Access Control) şi controlul legăturii logice (LLC – Logical Link Control).

Protocolul egal – la – egal (peer – to- peer) prevede trei posibilităţi de acces la mediu:

- Accesul multiplu cu sesizare de purtătoare şi detecţia coliziunii (CSMA/CD Carrier Sense Multiple Access with Collision Detection) IEEE 802.3
- Acces tip magistrală cu jeton (Token bus) IEEE 802.4. Aceasta este o variantă de interogare periodică – polling.

- Acces tip inel cu jeton (Token ring) IEEE 802.5. Şi aceasta este o variantă de interogare periodică – polling.

Protocoalele de comunicaţii tip LLC pentru transmisii punct la punct utilizează serviciul utilizează serviciul subnivelului MAC pentru a se oferi o transparenţă a aplicaţiilor faţă de mediul fizic. Există protocoale orientate pe caracter (byte) şi protocoale orientate spre bit.

Pentru transmisia asincronă se folosesc în mod obligatoriu caractere: 5 – 8 biţi pentru date, biţi de sincronizare şi controlul erorii. Dacă transmisia este sincronă datele pot folosii şiruri lungi de biţi sau octeţi.

Componentele reţelei între care se transmit date pot fi emiţător / receptor şi master / slave. Dacă masterul este emiţătorul iar componenta slave este receptorul atunci transferul informaţiei este de tip invitaţie la recepţie (selecting). Atunci când masterul este receptor iar componenta slave este emiţător se produce un transfer al informaţiei de tip invitaţie la emisie (polling).

O alternativă la metoda polling de transfer a informaţiei o constituie metoda difuzării aleatoare. În acest caz nu există o componentă master şi nu se pierde timp cu invitaţia la emisie. Mai multe staţii, de exemplu B, C, D, transmit în mod aleator pe acelaşi canal pachete de date de aceiaşi lungime spre staţia A.. Este posibil ca două sau mai multe pachete să se suprapună. În această situaţie se spune că a avut loc o coliziune şi pachetele se pierd. În mod curent, pentru fiecare pachet recepţionat staţia A transmite o confirmare pozitivă, în caz contrar nu transmite nimic şi pachetul se retransmite după o aşteptare de durată aleatoare. La metoda difuzării aleatoare nici o componentă a reţelei nu are sarcina transferului de date. Timpul câştigat datorită faptului că staţiile emit atunci când consideră necesar şi nu atunci când sunt invitate se poate pierde în cazul unor coliziuni repetate datorită retransmiterii datelor. Pe baza acestei metode s-a elaborat soluţia de transfer a informaţiei numită IEEE 802.3/Ethernet sau CSMA/CD (acces multiplu cu sesizare de purtătoare şi detecţia coliziunilor).

În cazul legăturilor punct – la – punct controlul transferului de informaţie este posibil în trei moduri:

1. legătură simetrică,
2. legătură asimetrică,
3. legătură echilibrată.

Legătura simetrică este formată din două sublegături distincte. În fiecare din cele două staţii care sunt în legătură sursa este master iar receptorul este slave, deci legătura este de tip selecting. Nu se foloseşte metoda polling. Acest tip de transmitere a informaţiei nu poate fi folosit în cazul legăturilor multipunct care sunt asimetrice prin definiţie: o staţie este master, celelalte sunt slave.

Legătura asimetrică consideră o staţie master şi alta slave. Se pot realiza legături multipunct.

Legătura echilibrată prevede că informaţia este de un singur tip, incluzând şi invitaţiile şi confirmările în acelaşi bloc mesaj (cadru).

9.3.4. Nivelele reţea, transport, sesiune, prezentare şi aplicaţie

- Nivelul 3, de rețea, precizează protocolul de stabilire, menținere şi terminare a conexiunii. Nivelele superioare sunt independente de tehnologia de transmisie a datelor şi de comutare.
- Nivelul 4, de transport, optimizează utilizarea serviciilor de rețea disponibile. Protocoalele definite la acest nivel asigură controlul fluxului şi recuperarea din erori pentru legătura cap – la –cap.
- Nivelul 5, de sesiune, determină cum se stabilesc, se administrează şi se termină conexiuni (sesiuni) între aplicații.
- Nivelul 6, de prezentare, asigură independența aplicațiilor față de reprezentările diferite ale datelor.
- Nivelul 7, de aplicație, asigură pentru utilizatori interfața cu mediul OSI. Elaborarea programelor de aplicație necesită cunoaşterea la nivel funcțional a protocoalelor de la nivelul fizic şi respectiv linie specifice diferitor aparate şi echipamente.

9.3.5. Protocolul PROFIBUS

Un exemplu de protocol la nivelul de automatizare este PROFIBUS – Germania. Acesta este compatibil cu cerințele modelului de referință ISO – OSI şi a fost însuşit de câteva firme importante din Europa: Siemens, Klockner Moller, Philips. Câteva caracteristici ale standardului PROFIBUS sunt următoarele:

- Standardul pentru nivelul fizic al modelului de referință ISO – OSI este RS – 485.
- Cuprinde trei tipuri de componente: master, slave şi repetor.
- Protocolul pentru nivelul linie al modelului de referință ISO – OSI este de tip polling pentru componentele slave şi de tip token pentru componente master.
- Celelalte nivele sunt de asemenea compatibile cu modelul de referință ISO – OSI
- Timpul de acces al unei componente a rețelei este de 100 ms.
- Semnalele sunt prelucrate autosincron în banda de bază cu următoarele debite:
    - 90 kbit/s pentru distanțe de 1200 m.
    - 187 kbit/s pentru distanțe de 600 m.
    - 500 kbit/s pentru distanțe de 200 m.
- Dimensiunea maximă a cadrului este de 255 octeți.
- Numărul maxim de componente a rețelei este de 32. Cu repetoare se poate ajunge la maximum 122 componente ale rețelei pe 5 tronsoane cu 3 repetoare.
- Structura rețelei este de tip magistrală.
- Mediul fizic de transmitere este format din două fire răsucite de lungime maximă 1200 m sau 4200 m cu repetoare.
- Rezistența terminală este de 120 ohmi.

9.3.6. Protocolul EIB Instabus

EIB (European Installation Bus) sau **Instabus**, are un protocol care stabileşte modul de conectare a componentelor. Câteva caracteristici.

- Transmiterea serială, asincronă a datelor pe bus (magistrală). Sincronizarea transmisiilor se realizează prin biţii de START şi STOP. Accesul la magistrală (BUS) se face prin metoda CSMA/CA (Carrier Sense – Multiple Access / Collision Avoidance) care elimină coliziunile telegramelor. Toţi participanţii la BUS primesc telegramele dar numai receptorii cărora le sunt adresate reacţionează. Dacă o componentă a sistemului vrea să emită o telegramă trebuie să urmărească magistrala până când nici o altă componentă nu mai emite (Carrier Sense). In situaţia în care BUS-ul este liber oricare componentă poate să iniţieze procedura de emitere (Multiple Acces). Dacă apar două emisiuni concomitente se va impune cel cu prioritate mai mare (Collision Avoidance)
- Alimentare la 24 V cc prin magistrală. Sursa de tensiune poate fi solicitată la 320 mA sau la 640 mA. şi este protejată la scurtcircuit. Sursele de alimentare debitează tensiunea de alimentare printr-o bobină de reactanţă care are rolul de a evita scurtcircuitarea telegramelor de date pe linia de BUS datorită sursei de alimentare.
- Două tipuri de obiecte: senzori şi elemente de execuţie. Fiecare componentă are o adresă fizică proprie. Exemplul simplu din primul capitol este un sistem instabus.
- Lungimea unei linii împreună cu toate ramificaţiile nu trebuie să depăşească 1000 m. Distanţa dintre o sursă de alimentare şi un participant la BUS trebuie să fie mai mică de 320 m. Pentru a evita coliziunile dintre telegrame trebuie ca distanţa dintre doi participanţi la BUS să fie mai mică de 700 m. Cablul de BUS poate fi montat paralel cu cablul de alimentare cu energie electrică a elementelor de execuţie fără să apară perturbări în transmiterea telegramelor.

Câteva protocoale folosite curent în SUA sunt prezentate în Tab 9.3

Tab 9.3

Protocolul	***Destinaţia***	***Controlat de***	***Autorul***
BACnet	Clădiri	Comitetul ASHRAE	Comitetul ASHRAE
DNP3	Producere de energie	Grupul utilizatorilor DNP3	Westronic
EIB – European Installation Bus	Clădiri	Siemens	Siemens
LonWorks	Clădiri	Echelon	Echelon
Modbus	Industrie	Schneider	Modicon
OPC – OLE for Process Control	Industrie	Fundaţia OPC	Microsoft
X10	Locuinţe	X10	X10

9.4. Sisteme de conducere în caz de pericol

Aceste sisteme sunt sisteme de domotică tipice şi sunt formate, în general, cu ajutorul unor automate programabile specializate. Sistemele moderne de conducere în caz de pericol conţin toate cele trei nivele ale sistemelor de domotică; management, automatizare şi câmp. Unele traductoare ale acestor sisteme sunt inteligente.

Cauzele care provoacă pericole în clădiri, grupuri de clădiri şi locuinţe sunt:

- Incendiul;
- Efracţia;
- Jaful;
- Avariile instalaţiilor tehnologice;
- Catastrofe naturale;
- Abateri de la regulile de convieţuire.

Echipamente clasice de conducere pentru aceste pericole sunt sistemele de securitate la efracţie şi sistemele de siguranţă la foc. Aceste sisteme sunt construite în jurul unor automate cu evenimente discrete care reacţionează la semnalele transmise de traductori specifici şi execută diferite acţiuni care să prevină pericolele şi să limiteze pierderile.

Sistemele de conducere automată în caz de pericol trebuie să realizeze următoarele acţiuni:

- Anticiparea prezenţei pericolului;
- Semnalizarea acustică şi vizuală (alarmarea) la nivel local, la nivel central şi la distanţă;

- ♦ Protecţie automată pentru limitarea pierderilor;
- ♦ Documentarea alarmelor în sensul explicării gradului de pericol, a locului în care au apărut şi a acţiunilor recomandate operatorului uman.
- ♦ Managementul alarmelor[200].

Pericolele se definesc în funcţie de scopul protecţiei:

- ✓ Viaţa;
- ✓ Bunuri materiale;
- ✓ Mediul natural înconjurător;
- ✓ Mediul social de convieţuire.

Standardele EN54 şi CEI 839 precizează că sistemul de conducere automată în cazul pericolelor se compune din:

1. Centrala ce conducere. Adeseori aceasta este desemnată sub denumirea de echipament de detecţie şi semnalizare sau centrală de alarmă. După cum am menţionat echipamentele moderne pot avea şi alte funcţiuni în afară de alarmare, cum ar fi protecţia sau documentarea.
2. Butoane de avertizare manuală.
3. Traductoarele de pericole. Aceste pot fi de tipul detectoarelor de avarie, a traductoarelor (monitoare analogice), sau a traductoarelor inteligente. Aceste dispozitive formează semnalele de intrare în centrală.
4. Dispozitive de alarmare, de protecţie automată sau de recepţie a semnalelor emise de centrală. Aceste dispozitive sunt conectate la ieşirea centralei.
5. Sursa de alimentare.

Sistemul de conducere în caz de pericol trebuie să respecte următoarele principii în funcţionarea sa:

a) Timpul de acţionare al sistemului de conducere trebuie să fie mai mic decât timpul minim în care sistemul protejat rezistă la pericol.
b) Fiabilitatea sistemului de conducere trebuie să fie mai mare decât a sistemului protejat.
c) Subordonarea sistemului de protejat faţă de sistemul de conducere trebuie asigurată numai în caz de necesitate. Sistemul de conducere trebuie să aibă capacitatea de a acţiona corespunzător limitării pierderilor numai în cazuri reale de necesitate. Nu este permisă această acţionare datorită unor efecte laterale a interacţiunii cu sistemul protejat, a defectării acestuia, sau a apariţiei unor defecte în însuşi sistemul de conducere.
d) Nefuncţionarea sistemului de conducere nu trebuie să împiedice funcţionarea sistemului protejat
e) Sistemul de conducere este subordonat operatorului uman. singurul care poate interpreta situaţii complexe periculoase şi care poate lua deciziile necesare.

[200] Gestion des alarmes în franceză, Alarm management în engleză.

Funcţionarea unui sistem de conducere în caz de pericol cuprinde mai multe etape distincte:

A. Prima şi cea mai importantă etapă constă în detecţia şi pericolelor. În această etapă sistemul de conducere nu trebuie să influenţeze în nici un fel activitatea sistemului protejat.
B. Identificarea pericolului şi verificarea identificării. Declanşarea alarmei. În această etapă operatorul uman poate influenţa decisiv evoluţia ulterioară a ansamblului sistem de conducere - sistem – protejat. Din această stare se poate trece imediat automat sau la intervenţia operatorului în starea următoare C de urgenţă – protejare. Dacă alarmă nu a fost reală se trece în prima stare A.
C. Etapa de urgenţă în care se execută toate acţiunile necesare limitării pierderilor datorate evoluţiei pericolului. Dacă pericolul persistă sistemul de conducere poate metode de supravieţuire prin restructurare şi poate subordona sistemul protejat. La terminarea pericolului se revine în etapa A.

Modul specific de funcţionare al sistemelor de conducere în caz de pericol a impus ca integrarea acestora în sistemele de domotică să se realizeze numai la nivelul superior de management. În felul acesta se respectă principiul subordonării faţă de operatorul uman şi se poate asigura prioritatea maximă a evenimentelor de pericol faţă de alte evenimente provocate de avarii tehnologice şi defecţiuni ale sistemului de conducere şi liniilor de transmisie[201].

9.4.1. Centrale de conducere în caz de pericol

Centrala de conducere recepţionează semnalele de intrare de la traductoare. Aceste semnale sunt condiţionate şi prelucrate pentru determinarea stării de pericol. În funcţie de rezultatele obţinute se elaborează diferite stări. Pentru acelaşi tip de supraveghere stările de alarmă corespunzătoare au aceiaşi prioritate. În caz contrar starea de alarmă corespunzătoare protecţiei vieţii are prioritate maximă. Semnalele transmise de butoanele acţionate manual au prioritate la prelucrare faţă de semnalele traductoarelor. Această atitudine reflectă concepţia că detecţia umană a pericolului este superioară celei automate. Starea de alarmă este transmisă sub forma unor semnale de ieşire pe trei căi diferite pentru:

- Dispozitivele de alarmare locală;
- Dispozitivele de recepţie a alarmelor;
- Dispozitivele de protecţie automată.

În afară de stările de alarmă există şi stările de defect ale sistemului. Aceste stări sunt transmise la distanţă pe o cale specială. Stările de alarmă au prioritate faţă de stările de defect iar transmisia semnalului de alarmă nu trebuie să fie blocată de starea de defect.

Traductoarele sunt situate, în general, la distanţă faţă de centrală. Conexiunea între ele se face prin linii de legătură. Acestea pot fi electrice, radio sau optice. Pentru liniile electrice defectele sunt următoarele:

---

[201] Protocoalele de comunicaţie standardizate pentru clădiri şi locuinţe cum ar fi BacNet sau EIB nu sunt potrivite pentru sistemele de conducere în caz de pericol.

- Linie întreruptă;
- Linie în scurtcircuit;
- Linie conectată la pământ.

9.4.2. Traductoare pentru sisteme de securitate la efracție

Cele mai folosite traductoare pentru sistemele de securitate sunt detectoarele pasive în infraroşu (PIR), monitoarele video de mişcare şi controlerele de acces. Sunt folosite în continuare traductoarele clasice de tip contact (buton de panică, senzor de poziție, etc.) şi a celor de vibrații sau şoc (detector de şoc, detector de geam spart). Acestea sunt foarte utile în special pentru semnalizarea alarmelor preventive.

9.4.3. Detectoare pasive în infraroşu - PIR

Orice corp uman emană o căldură în mediul înconjurător în spectrul infraroşu. Detectorul PIR este un senzor de proximitate bazat pe sesizarea variației termice captate în spectrul infraroşu de către un senzor piroelectric. Alarmele produse de către aceste detectoare nu au un caracter preventiv deoarece ele constată prezența în spațiul protejat şi nu încercarea de pătrundere în acesta. Aceiaşi observație este valabilă şi pentru monitoarele video.

Senzorul piroelectric este format dintr-un fotoelement şi un tranzistor TEC – MOS. Radiația în infraroşu determină fotoelementul să încarce electric capacitatea porții tranzistorului. Acesta va conduce un curent dependent de tensiunea aplicată pe poartă. Pentru a nu fi influențat de variațiile ambiante de radiație în infraroşu senzorul are o construcție diferențială conectată în opoziție. În acest mod valorile absolute ale iluminării ambiante sunt rejectate.

9.4.4. Monitoare video de mişcare

Aceste mai sun cunoscute şi sub denumirea de sisteme de televiziune cu circuit închis şi sunt singurele detectoare de efracție care furnizează la ieşire un semnal analogic în banda video stației centrale de urmărire. Imaginea este captată de către o matrice de fotodiode. Detectoarele video de mişcare sunt comparatoare între două imagini succesive ale senzorului cu fotodiode. Pentru efectuarea comparării una dintre imagini trebuie să fie memorată. Pornirea automată a sistemului de vizualizare şi/sau înregistrare video este dată de ale detectoare de mişcare volumetrice realizate cu ultrasunete, microunde sau PIR. Scanere video realizează compararea imaginii captată de către senzor cu cele existente într-o bancă de date. Datorită volumului mare de prelucrări ale imaginilor dinamice monitoarele video de mişcare sunt echipamente lente.

9.4.5. Controlere de acces

Controlerele de acces sunt echipamente electronice ataşate unor căi fizice de acces pentru a permite intrarea şi/sau ieşirea autorizată pentru zona protejată. Căile fizice de acces pot fi porți, uşi, lifturi automate de livrare a unui serviciu, etc. Autorizarea se poate face prin mai multe metode care diferențiază sistemele între ele. Una dintre metode foloseşte un cod de acces introdus cu ajutorul unei tastaturi. Un cod corect determină deblocarea accesului pentru un timp limitat.

Toate sistemele de control a accesului prezentate beneficiază de aportul unei baze de date referitoare la codurile persoanelor şi vehiculelor. După examinarea bazei de date se pot lua diferite acţiuni, de exemplu blocarea unor bariere. Din acest punct de vedere sunt nişte sisteme de gestionare a clădirilor, BMS. Se observă, pe de altă parte ca aceste sisteme au o structură ierarhică pe trei nivele şi din această cauză sunt şi sisteme de domotică.

9.4.6. Sisteme de identificare cu frecvenţă radio

Una din ultimele apariţii in lumea tehnologica actuala este RFID (Radio Frecquency IDentification) sau identificarea de proximitate.

RFID este o tehnologie avansata de colectare automata a datelor, aceasta tehnologie seamănă foarte mult cu sistemele bazate pe coduri de bare deoarece in mare au aceleaşi componente deşi au principii de funcţionare diferite .

Tehnologia codurilor de bare presupune un cititor optic (bazat pe raze laser) si o eticheta ce prezintă o succesiune de linii ataşata unui obiect. Spre deosebire de aceasta, RFID foloseşte un cititor (bazat pe un emiţător de frecvente radio in banda joasa) si un microcip (transponder sau tag) care poate fi implementat in interiorul obiectului sau pe o cartela ataşate de acesta.

Cititorul emite un câmp electromagnetic pe care tranponderul îl preia si prelucrându-l îl retrimite înapoi cititorului sub forma unor impulsuri, comunicarea intre tag si cititor făcându-se in fracţiuni de secunda.

Un sistem RFID este compus dintr-un cititor, un calculator si transponderul respective.

Datorita faptului ca nu este nevoie a se efectua un contact direct intre transponder si cititor acestea pot lucra in condiţii foarte vitrege (medii umede, uleioase, medii cu mult praf si mizerie) fiind foarte rezistente la temperaturi joase (40°C) si înalte (+200°C).

Tag-ul si cititorul pot fi separate de materiale textile sau medii nemetalice, transmiterea impulsurilor dintre ele efectuându-se prin acestea.

Cele mai simple aplicaţii ale detectării de proximitate se pot compara cu sistemele bazate pe coduri de bare, insa profitabilitatea acestui sistem se observa in aplicaţii precum GPS (Global Positioning Satelite system).

Alte aplicaţii pentru acesta tehnologie se bazează pe plasarea tag-ului in interiorul pneurilor auto, obţinându-se astfel informaţii rapide despre producător, data fabricaţiei, lotul de produse, locul comercializări etc sau implementarea tag-ului in cartele folosite la sistemele de acces in instituţii si restricţionarea accesului pe nivele de competenta.

9.4.7. Traductoare pentru sisteme de siguranţă la incendiu

Detectoarele automate de incendiu sunt elemente periferice ale instalaţiilor de semnalizare a incendiilor prin care se supraveghează in mod continuu sau la anumite intervale de timp un parametru fizic şi/sau chimic asociat incendiului.

In caz de incendiu, detectoarele declanşează un semnal care este transmis la centrala prin intermediul circuitelor de legătura.

Oricare ar fi tipul de detector, rolul sau intr-o instalaţie de semnalizare consta in a depista si semnaliza cat mai repede incendiul.

Pentru a acționa eficient, un detector automat de incendiu trebuie sa îndeplinească, in principal, următoarele caracteristici:

- ✓ funcționare sigura in condiții specifice de mediu (temperatura, umiditate, curenți de aer, concentrații de praf etc.)
- ✓ timp de răspuns rapid in prezenta parametrului supravegheat;
- ✓ stabilitate in timp a pragului de acționare;
- ✓ temporizare pentru eliminarea semnalizărilor false;
- ✓ imunitate la semnale perturbatoare;
- ✓ consum propriu redus de energie;
- ✓ semnalizarea stării0de buna funcționare (veghe);
- ✓ construcție simpla;
- ✓ întreținere si depanare uşoara.

Pentru a indica intrarea in stare de alarma, detectoarele de incendiu trebuie sa fie prevăzute cu semnalizare optica locala. Dispozitivele optice utilizate in acest scop trebuie sa emită lumina de culoare roşie, uşor vizibila de la distanta. Unele tipuri de detectoare mai au prevăzut suplimenta, fata de semnalizarea optica locala, un circuit separat de semnalizare la distanta. Acest circuit este necesar pentru punerea in funcțiune a unei lămpi de semnalizare montata intr-un loc uşor vizibil, pentru situațiile in care detectoarele sunt instalate in locuri greu accesibile.

In construcția detectoarelor, trebuie avut in vedere ca eventualele defecțiuni ale circuitelor, care pot scoate din funcțiune aparatul sau împiedica inițierea si transmiterea semnalului de incendiu la centrala, sa fie semnalizate local şi/sau la centrala, ca stare, de defect.

In general, utilizarea dispozitivelor electromecanice trebuie evitata. In cazuri speciale, când acest lucru este necesar, pentru transmiterea semnalului de incendiu, trebuie adoptata varianta cu contact normal închis in stare de veghe.

Soclurile necesare instalării detectoarelor trebuie sa aibă o singura poziție de fixare si sa fie cu contacte autocuratitoare. Sistemul de prindere a conductelor aferente circuitelor electrice trebuie sa asigure un contact electric sigur.

Detectoarele de incendiu se pot clasifica, in principal, după următoarele criterii:

- ✓ In funcție de parametrul supravegheat:

  a). detector de temperatura: sensibil la temperatura şi/sau gradient de temperatura şi/sau diferența de temperatura;

  b). detector de fum: sensibil la particulele produse de combustie şi/sau piroliza, suspendate in atmosfera:

  detector cu camere de ionizare: sensibil la particulele capabile sa afecteze curentul de ionizare

  detector optic: sensibil la particulele capabile sa afecteze absorbția sau împrăştierea radiațiilor din spectrul infraroşu şi/sau vizibil şi/sau ultraviolet;

  c). detector de gaze de combustie: sensibil la anumite produse gazoase rezultate in urma combustiei şi/sau descompunerii termice;

  d). detector de flacăra: sensibil la radiația electromagnetica emisa de flăcările de incendiu.

- ✓ In funcție de modul de răspuns la parametrul supravegheat:

  a). detector cu acționare statica: semnalizează la atingerea unei valori prestabilite a parametrului supravegheat;

b). detectoare cu acţionare diferenţiala: semnalizează la depăşirea unei valori prestabilite a diferenţei de mărime a parametrului supravegheat in cel puţin doua locuri;

c). detector cu acţiune velocimetrică: semnalizează la depăşirea unei valori prestabilite a vitezei de creştere (gradient) a parametrului supravegheat.

- ✓ In funcţie de configuraţia senzorului:

  a). detector punctual: acţionează la parametru supravegheat din vecinătatea unui senzor punctual;

  b). detector multipunctual: acţionează la parametrul supravegheat din vecinătatea mai multor puncte;

  c). detector liniar: acţionează la parametrul supravegheat din vecinătatea unei linii continue.

- ✓ In funcţie de modul de reutilizare după acţionare (producere alarma, de verificare):

  a). detector reutilizabil: poate fi readus in stare de funcţionare, in vederea unei noi acţionari, după încetarea condiţiilor care au produs acţionarea sa, fără înlocuirea vreunei componente;

  b). detector parţial reutilizabil: poate fi readus in stare de funcţionare, după încetarea condiţiilor care au produs acţionarea sa, prin înlocuirea unor componente;

  c). detector nereutilizabil: care nu mai poate fi reutilizat după acţionare, fiind necesara înlocuirea sa.

Clasificările de mai sus nu sunt limitative, putând exista si diverse combinaţii ale tipurilor prezentate sau funcţionând pe alte principii sau alte criterii de clasificare.

9.4.8. Butoane manuale de semnalizare

Butonul manual de semnalizare reprezintă dispozitivul prin intermediul căruia se poate semnaliza manual, de către om, apariţia unui incendiu. Cu toata răspândirea din ce in ce mai mare a detectoarelor automate de incendiu in instalaţiile de semnalizare, butoanele manuale de semnalizare sunt folosite, încă, pe scara larga, deoarece prezintă o construcţie simpla si siguranţa ridicata in exploatare. Folosirea butoanelor manuale in cadrul instalaţiilor automate de semnalizare a incendiilor este justificata si prin faptul ca, in anumite situaţii, incendiul poate fi observat de către om înainte de declanşarea unui detector automat si ca atare este raţional ca instalaţiile sa se prevadă si cu aceasta posibilitate.

Instalaţiile de semnalizare a incendiului se prevăd numai cu acţionare manuala doar in acele situaţii in care intervenţia pentru stingerea in caz de incendiu se asigura in timp util.

Butoanele de semnalizare a incendiilor se vor amplasa in locuri vizibile, uşor accesibile, de preferinţa lângă uşa, la intrarea in casa scărilor sau in aceasta si in general in punctual de circulaţie obligatorie in caz de evacuare. In cazul spatiilor cu suprafeţe mari de supraveghere (încăperi, culoare, hale de producţie etc.), butoanele de semnalizare se vor amplasa astfel încât nici o persoana sa nu aibă nevoie a se deplasa mai mult de circa 50 m, de la orice poziţie din clădire, spre a da alarma de incendiu. Butoanele de semnalizare se amplasează de regula la o

înălțime de circa 1,4 m de la pardoseala. Atunci când este necesar, locul de amplasare al acestora va fi iluminat corespunzător pentru a fi uşor observata. Pentru clădirile cu mai multe nivele, butoanele de semnalizare se vor amplasa la fiecare nivel, in apropierea scărilor sau a altor cai de acces. Nu este admisa conectarea butoanelor de semnalizare, de pe diferite nivele, la acelaşi circuit de linie din centrala de semnalizare.

Spatiile in care se prevăd detectoare automate, conform legislației in vigoare, vor fi dotate in mod obligatoriu si cu butoane manuale de semnalizare, instalate pe circuite de linii distincte. La baza acestei prevederi, a stat considerentul realizării unei siguranțe ridicate in semnalizarea apariției unui incendiu prin semnalizare manuala, de către om, înainte de acționarea unui detector automat de incendiu, pe de o parte, cat si existenta unei rezervări in cazul in care circuitul de linie cu detector automat ar fi defect, pe de alta parte.

Principiul care sta la baza funcționarii butoanelor de semnalizare manuala este mecanic si consta, in funcție de varianta constructiva a aparatului, in închiderea sau deschiderea unor contacte. Datorita siguranței mai ridicate in transmiterea semnalizării de incendiu, in instalațiile de semnalizare, se utilizează cu precădere butoanele de semnalizare care – in starea normala de veghe – prezintă un contact normal închis (CNI) si in alarma – contact normal deschis (CND). Aceasta cerința a rezultat din practica, unde s-a constatat ca – datorita unei întreținери defectuoase in special a elementelor de etanşeizare in locurile cu mult praf, umezeala, substanțe corozive etc. – transmiterea semnalizării de incendiu nu s-a mai putut face la acționarea butonului, datorita oxidării sau depunerii prafului pe contactele din interiorul aparatului.

La unele tipuri de butoane de semnalizare, exista si posibilitatea realizării unei legături fonice cu centrala de semnalizare, legătura ce se stabileşte in mod automat după acționarea acestuia pentru transmiterea semnalizării de incendiu.

Aceste tipuri sunt deosebit de utile, întrucât operatorul de serviciu se poate informa cu date privind natura si amploarea incendiului.

Din punct de vedere al construcției, butoanele de semnalizare se fabrica in variantele:

pentru medii normale, destinate amplasării in interiorul construcțiilor sau in exteriorul acestora;

pentru medii explozive (de interior si de exterior);

pentru mediu naval (de interior si exterior).

9.4.9. Detectoare de temperatură

Cele mai răspândite detectoare de incendiu aflate in exploatare in instalațiile de semnalizare a incendiilor sunt detectoarele termice. Acest fapt se datorează in primul rând simplității, robusteții si prețului relativ scăzut al aparatelor. Cu toate acestea, detectoarele care funcționează cu fir sau aliaj fuzibil au si anumite inconveniente care nu pot fi neglijate.

Dintre acestea, cele mai importante sunt:

- inerția termica a aliajului nu asigura sesizarea destul de rapida a apariției incendiului, in special daca gazele fierbinți nu se propaga direct spre detector;

- aria de supraveghere este redusa, ceea ce face sa fie necesara montarea unui număr mare de detectoare;
- temperatura mediului ambiant influenţează timpul de răspuns.

Detectoarele de temperatura cu elemente de bimetal cunosc, de asemenea, o mare răspândire. Funcţionarea acestora se bazează pe proprietatea dilatării inegale a doua metale cu coeficienţi diferiţi de dilatare (de exemplu: alama) sudate pe suprafaţa. Prin încălzirea lamei, datorita dilatării inegale a celor doua fete, aceasta se va curba in direcţia lamei cu coeficient de dilatare mai mic, si, in funcţie de tipul constructiv, se va închide sau deschide un contact electric.

Spre deosebire de detectoarele cu fuzibil, la cele cu bimetal temperatura de declanşare poate fi reglata in limite foarte largi.

Pentru a elimina dezavantajele legate de influenta temperaturii iniţiale a mediului asupra detectorului, se utilizează detectoarele diferenţiate si velocimetrice. Indiferent de principiul de funcţionare, detectorul termodiferential iniţiază un semnal de alarma atunci când diferenţele, normal mici, de temperatura, in doua sau mai multe locuri, depăşesc – pentru o durata de timp suficienta – o anumita valoare prestabilita.

Detectoarele termovelocimetrice funcţionează pe principiul măsurării vitezei de creştere a temperaturii in unitatea de timp.

Sensibilitatea acestor aparate este de ordinul a 5-8 °C/minut. Detectoarele termodiferentiale si termovelocimetrice sunt mai sensibile si funcţionează mai rapid decât cele termostatice (de maxim). In cele mai multe situaţii, aceste tipuri de detectoare sunt combinate si cu funcţiunea de semnalizare termostatică (de maxim).

Detectoarele pneumatice de temperatura se compun, in principiu, din doua camere de aer suprapuse. Una din camere (compensare) este izolata fata de atmosfera înconjurătoare. Cealaltă camera (receptoare0 este in legătura directa cu atmosfera. Intre cele doua camere, se afla o membrana metalica elastica care formează primul electrod. In stare normala, membrana se afla in contact cu un vârf metalic prevăzut cu un şurub de reglaj, care formează cel de-al doilea electrod.

Când temperatura mediului creste încet, presiunea aerului in cele doua camere se echilibrează prin intermediul ajutajului care face legătura intre acestea.

In caz de incendiu, temperatura creste brusc si – implicit – membrana este deplasata înspre camera de compensare, întrerupând contactul cu cel de-al doilea electrod.

Si aceste tipuri de detectoare, in cele mai multe cazuri, sunt asociate cu funcţia termostatică.

Detectoarele de temperatura cu aliaj fuzibil, bimetal si pneumatic fac parte din categoria detectoarelor care nu consuma energie electrica in funcţionare.

Pentru a elimina dezavantajele legate de inerţia termica relativ mare a senzorilor prezentaţi anterior, s-au construit detectoare cu elemente semiconductoare. Dintre acestea, cel mai uzual sunt folosite termorezistenţele si termistorii. De regula, aceste detectoare îndeplinesc o funcţie dubla, termostatică si termovelocimetrica.

Funcţionarea acestor tipuri de detectoare se bazează pe variaţia rezistentei senzorilor sub influenta temperaturii. Variaţia de rezistenta este prelucrata de

circuite electronice si in final transmisa sub forma de semnal de alarma la centrala de semnalizare.

9.4.10. Detectoare de fum cu ionizare

O camera de ionizare tipica consta in doua placi încărcate electric si o sursa radioactiva (in mod obişnuit Americiu 241) pentru ionizarea aerului dintre placi. Sursa radioactiva emite particule care intra in coliziune cu moleculele de aer dislocându-le electronii. O parte din moleculele de aer devin ioni pozitivi prin pierderea electronilor. O alta parte a moleculelor primeşte electroni devenind astfel ioni negativi. Se creează astfel un număr egal de ioni pozitivi si negativi. Ionii pozitivi sunt atraşi de placa negativa, iar cei negativi de placa pozitiva. Acest fenomen generează un curent de ionizare, care poate fi măsurat prin circuitul electronic conectat la cele doua placi.

Particulele rezultate din combustie sunt mult mai mari decât moleculele de aer ionizate. La intrarea particulelor de combustie in camera de ionizare, acestea vor intra in coliziune cu moleculele de aer ionizate si se vor combina cu acestea.

Unele particule astfel rezultate vor fi pozitive, iar altele vor fi negative. Aceste particule relativ mari continua sa se combine cu alţi ioni, devin centre de recombinare si totalul particulelor ionizate din cameră se reduce. Reducerea numărului de particule ionizate conduce la scăderea curentului de ionizare mai sus menţionat, care este permanent monitorizat. La scăderea acestui curent sub o valoare predeterminata, se depăşeşte un prag dincolo de care este stabilita condiţia de alarma.

Schimbările in presiunea si umiditatea atmosferica pot afecta curentul de ionizare si pot crea un efect similar aceluia de pătrundere a particulelor de combustie. Pentru compensarea efectelor nedorite generate de schimbările de presiune si umiditate a apărut camera de ionizare dual㠄, care se foloseşte pe scara larga pe piaţa detectoarelor de fum.

Un detector cu camera dual㠄 foloseşte doua camere de ionizare. Una dintre ele este camera de detecţie care este deschisa către aerul din exterior,. Camera de detecţie este afectata de microparticule, de presiunea si umiditatea atmosferica. Cea de-a doua camera serveşte ca referinţa si este parţial închisa fata de aerul din exterior. Ea este afectata numai de presiunea si umiditatea atmosferica întrucât deschiderile mici prin care pătrunde aerul nu permit intrarea microparticulelor cum ar fi particulele de fum. Circuitul electronic monitorizează ambele camere si compara semnalele.

Schimbările de presiune sau umiditate ale atmosferei afectează in mod identic semnalele de ieşire ale ambelor camere de ionizare, care in acest fel se anulează reciproc. La intrarea particulelor de fum in camera de detecţie scade curentul de ionizare al acesteia comparativ cu curentul din camera de referinţa, care virtual rămâne neschimbat. Diferenţa de curent rezultata este detectata de circuitul electronic.

Exista totuşi o serie de probleme care pot afecta detectoarele cu camera dual㠄 de ionizare: praful, umiditatea excesiva (condensul), curenţi de aer semnificativi, mici insecte care pot fi "citite" ca particule de combustie de către circuitul electronic al detectorului. Cu cat detectorul este calibrat mai sensibil cu

atât mai mult aceste probleme pot afecta performantele detectorului generând alarme false.

9.4.11. Detectoare de fum optice

La aceste detectoare fumul afectează intensitatea unui fascicol de lumina ce trece prin aer. Fumul poate obtura sau chiar bloca acest fascicol. De asemenea poate cauza difuzia luminii datorita reflexiilor pe particulele de fum. Detectoarele de fum fotoelectrice sunt proiectate deci pentru a sesiza fumul folosind aceste efecte ale fumului asupra luminii.

9.4.12. Detectoare de fum fotoelectrice cu obturarea luminii

Unul dintre tipurile de baza de detector de fum fotoelectric este detectorul cu obturarea luminii. Acesta este format dintr-o sursa de lumina si un dispozitiv fotosensibil, cum ar fi o fotodioda. Semnalul de ieşire al elementului fotosensibil este afectat de particulele de fum care blochează parţial fascicolul. Schimbarea acestui semnal este sesizata de circuitul electronic al detectorului si daca se depăşeşte un anumit prag se generează semnal de alarma. Detectoarele cu obturare sunt de obicei de tip cu fascicul proiectat, la care sursa de lumina extinde aria de protejat.

9.4.13. Detectoare de fum fotoelectrice cu difuzarea luminii

Majoritatea detectoarelor de fum sunt de tip focalizat si operează pe principiul luminii difuzate. O dioda LED luminează o arie care uzual nu este "văzuta" de elementul fotosensibil, care de obicei este o fotodioda. La pătrunderea particulelor in calea luminii, aceasta este reflectata pe elementul fotosensibil, activând detectorul.

9.4.14. Sistemul MicroSAM

Micro Supervizorul Adresabil Modular MicroSAM, produs în România de AUTOMATICA SA, este un *sistem de tip adresabil* destinat supravegherii, detectării şi alarmării în cazul apariţiei unor pericole în clădiri şi locuinţe.

Sistemul este compus dintr-o centrală de semnalizare şi dispozitive periferice adresabile (butoane, detectoare de fum, de flacără, de temperatură, sonerii, etc.). Centrala permite realizarea a 4 linii (circuite) de semnalizare şi pe fiecare dintre acestea se pot stabili 15 adrese distincte. Fiecare circuit de semnalizare permite conectarea atât a elementelor de iniţiere a alarmei cât şi a elementelor de execuţie (avertizare).

Indicarea evenimentelor care au loc în sistem se efectuează local pe un afişor cu patru caractere sintetizate prin şapte segmente, aparţinând centralei, sau prin tipărire la o imprimantă conectată pe ieşirea serială RS-232 a centralei. Indicarea evenimentelor produse în sistemul de supraveghere se poate efectua şi la distanţă, datorită comunicaţiei care se poate stabili între centrala MicroSAM şi un calculator la nivelul ierarhic superior. De asemenea, la distanţă se poate comunica automat prin linie telefonică, folosind principiul comutării circuitelor, dispozitivul de apel telefonic având o capacitate de maxim patru numere (abonaţi).

Centrala MicroSAM transmite interfețelor adresabile mai multe comenzi: SINCRO, APEL, CITIRE si COMANDA. Primele doua comenzi utilizează modularea duratei de suprapunere peste tensiunea continua de alimentare a unei purtătoare de frecventa fixa generata de centrala. Ultimele doua semnale permit citirea informației de la interfață, respectiv transmiterea unei comenzi (active) către aceasta, ambele utilizând variația duratei lipsei purtătoarei susmențională. Ciclul de funcționare debutează cu alimentarea interfețelor, urmat de transmiterea de către MicroSAM a unui semnal de sincronizare SINCRO.

Fiecare interfața care recepționează acest semnal, va încărca in comparatorul de adrese, adresa proprie, prestabilita, a acesteia. Recepționarea succesiva in continuare a mesajelor APEL de către toate interfețele conectate pe linie va determina ca un singur comparator de adrese sa dea un semnal de egalitate si in acest mod, interfața va putea fi CITITA de către Centrala MicroSAM. Interfața răspunde printr-un impuls de curent de durata variabila, dependenta de starea sa. Centrala interpretează acest semnal si decide daca se reia ciclul cu un alt semnal APEL transmis pe linie sau trimite comanda de acționare a ieşirii.

In starea de veghe a detectorului clasic, conectat la interfață aceasta răspunde printr-un puls de curent de durata corespunzătoare absorbit de detector si rezistenta cap de linie a lui. In starea de alarmare a detectorului, curentul acestuia va creste, crescând si durata răspunsului către MicroSAM.

Verificarea corectitudinii declanşării detectorului se realizează de către centrala printr-o COMANDA, care determina interfața sa întrerupă alimentarea detectorului. Nedeclanşarea imediata a acestuia, permite centralei sa rejecteze alarmele false. De asemenea, netransmiterea pulsului de curent la momentul interogării interfeței indica centralei o defecțiune apărută in aceasta.

Dispozitivele de ieşire sunt in principiu rele bistabile care primesc comenzi de RESET la fiecare CITIRE si SET la fiecare COMANDA.

Sistemul ierarhizat pentru detectare şi alarmare la incendii a fost instalat, pus în funcțiune şi experimentat la laboratorul de automatizări al Facultății de Instalații din Bucureşti. S-a pornit de la echipamentele specializate aflate în producția curentă la AUTOMATICA SA din Bucureşti în anul 1997 şi anume: centrala de semnalizare şi alarmare MicroSAM, un detector de fum, un detector de flacără, un detector de temperatură ridicată şi o interfață specializată pentru conectarea detectoarelor de incendiu la centrală.

Centrala de semnalizare a fost conectată în vederea comunicației cu un calculator PC, aflat la nivelul ierarhic superior. Comunicația între centrală şi calculator se efectuează serial, cu o viteză de 2400 bauds, este de tip asincron şi respectă standardul RS-232C.

Sensul transmisiei informației este de la centrala MicroSAM, aflată la nivelul automatizare al sistemului ierarhizat, către calculatorul situat la nivelul supraveghere/gestiune. Este posibilă transmisia informației şi în sens invers, atunci când centrala se programează de la nivelul calculatorului.

Distanța maximă dintre centrala MicroSAM şi calculator poate fi de 15 m, atunci când comunicația se realizează conform standardului RS-232C. Dacă situația concretă dintr-o clădire impune comunicația la distanțe mai mari, atunci se poate utiliza cea de-a doua ieşire serială a centralei, care foloseşte standardul RS-

485. La capătul dinspre calculator al cablului de comunicaţie se va utiliza un adaptor RS-232 / RS-485 şi distanţa maximă poate fi de 1200 m.

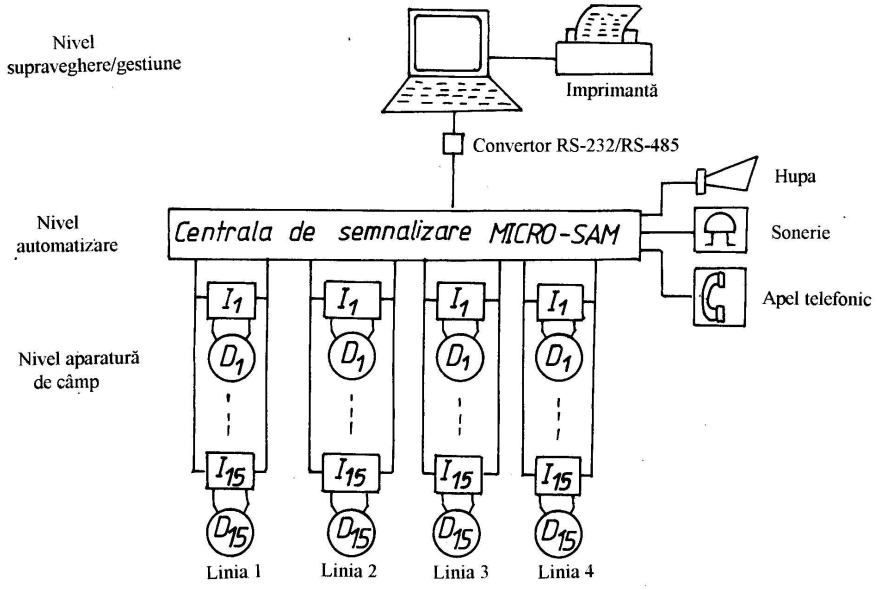


Fig. 9.8 Structura sistemului ierarhizat realizat cu central MicroSAM.

Caracterele transmise de centrală şi recepţionate de calculator se obţin prin adăugarea la cei opt biţi informaţionali (codul ASCII al caracterului transmis) a informaţiei cadru, constituită dintr-un bit de START şi un bit de STOP. Sincronizarea emiţătorului cu receptorul are loc numai pe durata transmisiei unui caracter.

Aceeaşi ieşire serială a centralei de semnalizare poate fi conectată şi la o imprimantă, unde se tipăresc rapoarte despre starea sistemului de supraveghere şi alarmare la incendii, în momentele semnificative de timp.

În regim de monitorizare, centrala MicroSAM transmite la calculator sau tipăreşte la imprimantă orice eveniment care se produce în sistem: alarme provenite de la butoane sau detectoare, întreruperi sau scurtcircuite în liniile de semnalizare cablate în clădirile sau spaţiile supravegheate, întreruperea alimentării centralei de la reţeaua de 220V c.a., descărcarea acumulatoarelor, etc.

Un sistem ierarhizat pentru detectare şi alarmare la incendii a fost realizat cu MicroSAM Fig. 9.8. Calculatorul de la nivelul supraveghere/gestiune este acelaşi calculator care asigură nivelul ierarhic superior în sistemul de conducere al instalaţiei de încălzire şi în sistemul pentru asigurarea şi monitorizarea altor servicii din clădire.

În Fig. 9.8 s-au folosit următoarele notaţii:

- I_1, I_2,...,I_{15} interfeţele adresabile ale detectoarelor de incendiu;
- D_1, D_2,...,D_{15} detectoarele de incendiu.

Dacă dimensiunile sau alte particularităţi ale clădirilor supravegheate la incendii, impun utilizarea a mai mult de patru linii de semnalizare cu un total de butoane şi detectoare mai mare de 60 elemente, atunci se poate opta pentru soluţii cu mai multe centrale conectate în reţea. Se accentuează în acest mod caracterul distribuit al sistemului de supraveghere şi alarmare la incendii. Centrala MicroSAM nu permite acest mod de funcţionare.

9.5. Sisteme de echipamente

Elementele sistemului pot fi diferite echipamente interconectate: motoare electrice, pompe, cazane, rezervoare, aparate de măsurare şi automatizare, robinete de reglare, etc. Un model al sistemului de echipamente este schema tehnologică cu aparatura de automatizare[202]. Un exemplu este prezentat în Fig. 9.9.

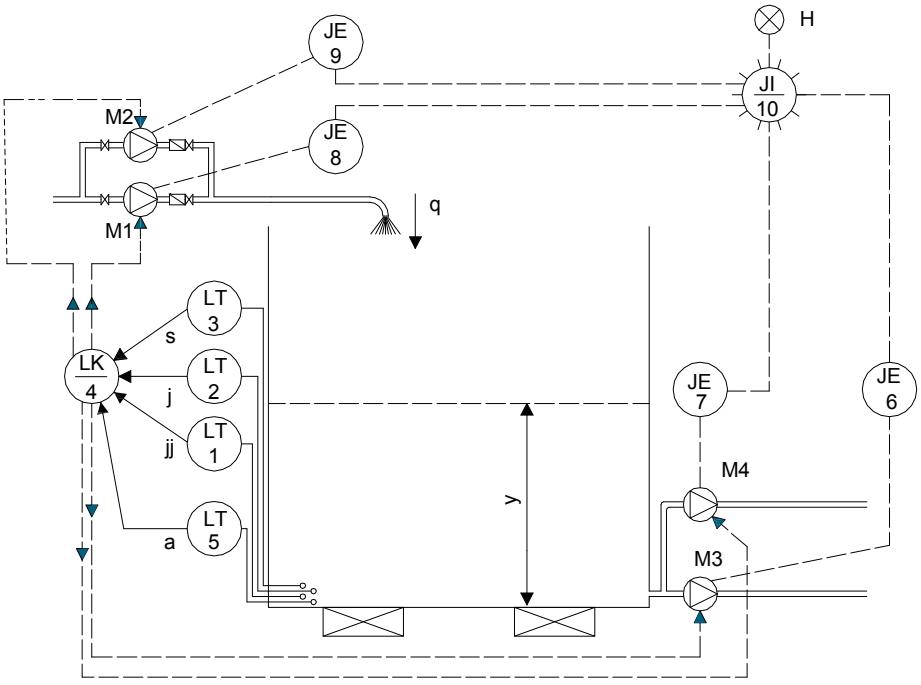


Fig. 9.9 Schema tehnologică a unei instalaţii pompă – rezervor.

Reprezentarea echipamentelor de automatizare şi măsurare este standardizată, câteva dintre simboluri şi semnificaţia lor fiind prezentată în Tab 9.4, Tab 9.5 şi Tab 9.6

Tab 9.4 Simboluri pentru echipamente

	Accesibile	Montate în câmp	Montate în

[202] P&ID – Pipe and Instrumentation Diagram, conform standardului ANSI/ISA S 5.1 – 1984 (R 1992)

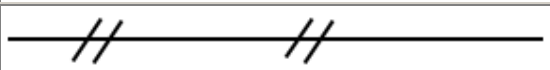
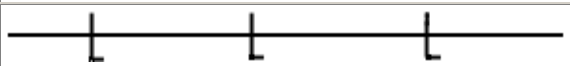
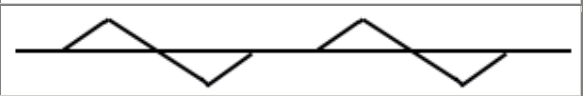
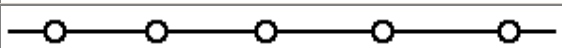
	operatorului		locații auxiliare
Echipamente discrete	1 	2 	3 
Echipamente cu display	4 	5 	6 
Calculatoare	7 	8 	9 
Automate programabile	10 	11 	12 

Tab 9.5 Semnificația etichetei simbolului echipamentului

Litera	Prima literă: variabila măsurată sau reglată	A doua literă: funcția echipamentului
A	Analiză	Alarmă
B	Ardere	La alegere
C	La alegere	
D	La alegere	
E	Tensiune	Sensor (element primar)
F	Debit	
G	La alegere	
H	Manual	
I	Current electric	Indicare
J	Putere	
K	Timp	
L	Nivel	Lumină
M	La alegere	
N	La alegere	La alegere
O	La alegere	
P	Presiune	

Q	Cantitate	
R	Radiație	Înregistrare
S	Viteză, Frecvență	
T	Temperatură	
U	Multivariabile	Multifuncții
V	Vibrații	
W	Weight, force	
X	Neclasificat	Neclasificat
Y	Eveniment, stare	
Z	Poziție, dimensiune	

Tab 9.6 Tipuri de conexiuni între echipamente

Tipul liniei	Linia de conexiune
Conectare la proces sau alimentarea echipamentului	
Semnal pneumatic:	
Semnal electric	
Tub capilar	
Semnal hidraulic	
Semnal acustic	
Magistrală (bus)	

Un alt model pentru sistemele de echipamente îl reprezintă schemele desfăşurate electrice, pneumatice sau hidraulice. Un exemplu de schemă desfăşurată electrică este prezentat în Fig. 9.10.Alte exemple foarte simple de scheme desfăşurate electrice s-au folosit în capitolele dedicate sistemelor logice combinaționale ş sistemelor cu evenimente discrete care folosesc contacte, relee şi automate programabile logice.

Mediile moderne de dezvoltare a aplicațiilor sistemice, cum este mediul Dynast prezentat în ultimul capitol, folosesc pentru sistemele de echipamente modele complexe modele (scheme) tehnologice, modele funcționale (scheme bloc) şi modele structurale (rețele de dipoli fizici). În exemplul din *Fig. 9.11* motorul electric şi instrumentele de măsură fac parte dintr-o schemă tehnologică,

regulatorul PID şi integratorul formează o schemă bloc (funcţională) iar sursa de tensiune V este elementul unei reţele electrice (model structural).

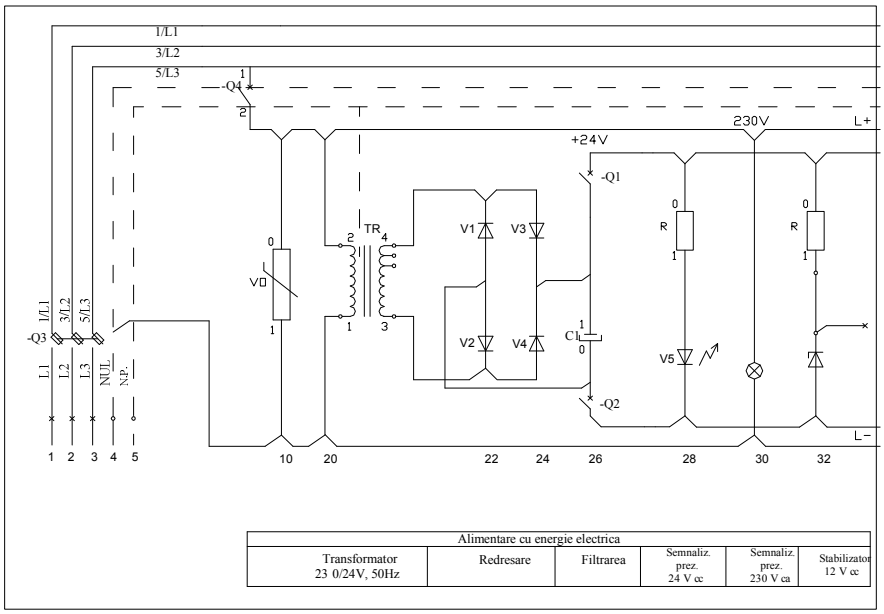


Fig. 9.10 Modelul unui sistem de echipamente sub formă de schemă desfăşurată electrică

Dynast permite simularea sistemului de echipamente (motor, regulator şi aparate de măsură) descris de modelul combinat din Fig. 9.11.

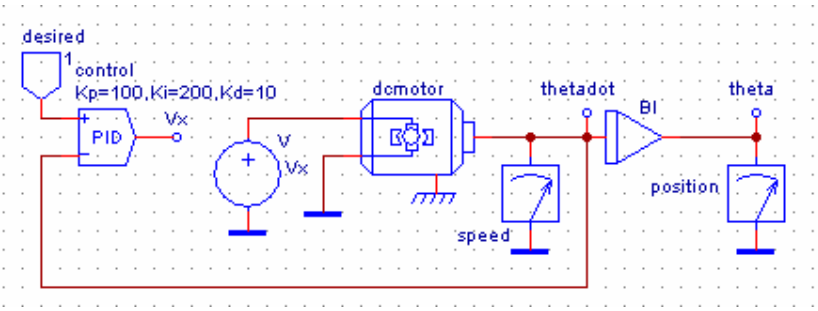


Fig. 9.11 Modelul unui sistem de echipamente pregătit pentru simulare cu Dynast

10. Medii de dezvoltare a aplicaţiilor

Modelarea, analiza şi simularea sistemelor se face adeseori asistată de calculator. Câteva dintre programele folosite pentru sistemele fizice şi sistemele de conducere automată sunt Dynast, KitSAS şi Scilab.

10.1. KitSAS[203]

Kitul de Simulare şi Analiză a Sistemelor – **KitSAS**, versiunea 6.12 destinată sistemelor de operare Win95, Win98 şi WinXP, este un mediu de dezvoltare şi execuţie a programelor destinate construirii, simulării si analizei sistemelor dinamice, incluzând atât sub-sistemele continue cât şi cele discrete. KitSAS include mai multe editoare pentru construirea modelelor prin interconectarea elementelor sau blocurilor. Elementele reprezintă funcţii de bază predefinite sau funcţii definite de utilizator. Modelul construit apare sub forma unui fişier text care poate fi compilat şi executat. Kitul conţine şi o bibliotecă de peste 300 blocuri şi sisteme electrice şi neelectrice care pot constituii un punct de pornire pentru construirea modelelor unor noi sisteme.

Conceptele, metodele şi terminologia kitului se bazează pe acţiunile inginereşti efectuate în mod obişnuit atunci când se studiază un obiect tehnic. Deoarece metoda de analiză inginerească cea mai utilizată este analiza experimentală, kitul încearcă să o reproducă prin simulare. In acest caz variabila independentă este timpul. Metode experimentale mai elaborate, îndeosebi pentru sistemele electrice şi sistemele automate, folosesc analiza în domeniul frecvenţă pentru care variabila independentă este frecvenţa. Poate şi mai des în activitatea inginerească se face analiza experimentală in regim staţionar. Kitul simulează şi aceasta acţiune pentru circuitele electrice şi electronice sau sistemele discrete logice.

KitSAS poate fi imaginat ca o cutie cu scule si obiecte necesare simulării şi analizei sistemelor. Printre obiectele şi sculele cele mai folosite ale kitului menţionăm elementul, blocul, standul experimental, panoul de comandă, înregistratorul numeric şi grafic al rezultatelor, osciloscopul cu două canale, instrumentul de măsură indicator, generatorul manual de semnal şi analizatorul de frecvenţă.

KitSAS poate fi prezentat şi altfel, prin analogie cu limbajele de programare şi mediile lor vizuale de dezvoltare rapidă a aplicaţiilor. Din acest punct de vedere KitSAS este un limbaj neprocedural, orientat spre obiecte şi cu structura variabilă a programului.

Dintre produsele comerciale, programul Simulink scris în Matlab şi programul Scicos elaborat în Scilab sunt foarte apropiate de componenta de simulare a kitului. Programul Pspice este foarte asemănător cu componenta kitului pentru analiza în regim sinusoidal a circuitelor electrice şi electronice. În toate

[203] Un exemplu de aplicare în paragraful 3.3

cazurile menţionate deosebirile se găsesc în special la interfeţele grafice de intrare pentru construirea modelelor.

O versiune DOS a KitSAS se găseşte în subdirectorul Utile. Programele existente în Magazie sunt valabile atât pentru versiunea DOS cât şi pentru versiunea Windows a programului KitSAS.

O prezentare amănunţită a kitului şi al modulului său de programare se găseşte în lucrarea: Sorin Larionescu, KitSAS – Simularea şi Analiza Sistemelor, Ed. MATRIX ROM, Bucureşti, 1999. Versiunea electronică a manualului sub forma unui fişier pdf se găseşte în directorul DocDos al directorului în care se instalează kitul..

Pentru o introducere simplă şi rapidă în modul de folosire şi programare al kitului de simulare se consideră experimentul ingineresc efectuat asupra unei instalaţii de umplere a unui rezervor cu ajutorul unei pompe. Schema tehnologică este prezentată în Fig. 10.1.

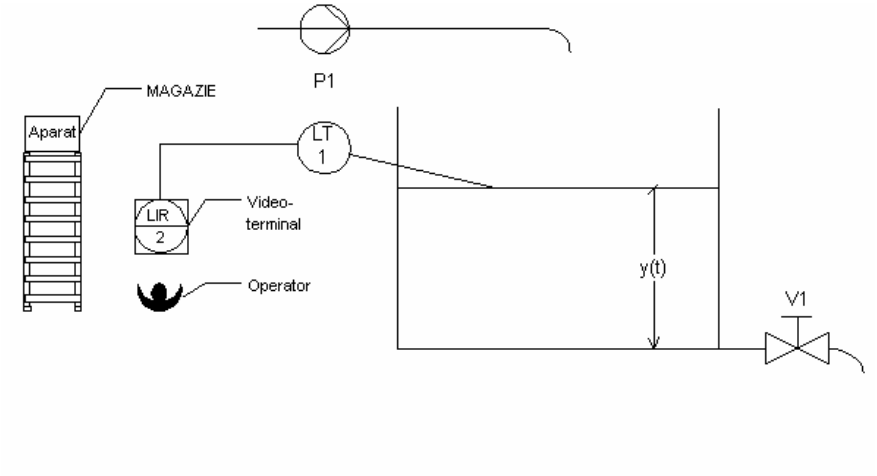


Fig. 10.1 Sistemul pompă-rezervor

Se deosebesc două componente principale ale sistemului: a) instalaţia tehnologică şi b) echipamentul de măsurare şi conducere.

a). *Instalaţia tehnologică.* In partea tehnologică sistemul este format din pompa P1 de umplere şi rezervorul împreună cu robinetul V1. Studiul tehnic urmăreşte determinarea modului de variaţie în timp a nivelului în rezervor y(t) pentru diferite regimuri de funcţionare a pompei P1. Se observă imediat că nivelul y(t) se modifică diferit în funcţie de starea robinetului V1, închis sau deschis. O primă soluţie pentru realizarea studiului tehnic este metoda experimentală: se porneşte pompa P1 şi se măsoară nivelul y(t).

b*). Echipamentul de măsurare şi conducere*. Aparatele folosite se găsesc situate în trei locuri: pe instalaţia tehnologică, în camera sau panoul cu aparate active şi în magazie sau dulapul cu aparate inactive. Aceste aparate sunt de obicei electric, electronice, electro-mecanice sau fluidice şi pot fi de tip analogic sau numeric. In figură sunt prezentate următoarele aparate *active:* traductorul de nivel 1 cu interfaţa pentru transmiterea la distanţă şi panoul de conducere format dintr-un calculator împreună cu videoterminalul 2. Operatorul poate lua alte aparate,

plăci sau blocuri din Magazie şi le poate monta pe instalaţia tehnologică sau panoul de conducere. În felul acesta întreg sistemul experimental dobândeşte funcţiuni noi. Operatorul şi calculatorul se găsesc la distanţă faţă de instalaţia tehnologică.

Experimentul decurge în modul următor: operatorul porneşte pompa şi calculatorul înregistrează variaţia nivelului y(t). Apare totuşi o problemă, intervalul de timp sau pasul cu care se înregistrează rezultatele. Cu cât pasul este mai mic cu atât este mai bine. In mod curent calculatorul poate realiza o măsurare la fiecare fracţiune de secundă. Dar procesul de umplere al rezervorului este lent, poate dura zeci de minute, deci mii de secunde. Rezultă mii de rezultate care nu pot fi interpretate de operator. Calculatorul trebuie să realizeze interfaţa cu operatorul prin analiza rezultatelor si prezentarea lor sub formă tabelară, grafică sau sub forma unor performanţe standard.

Instalaţia tehnologică poate fi modelată simplu, prin mai multe elemente conectate în serie ca în Fig. 10.2. Primul şi al doilea element modelează pompa, al treilea şi al patrulea rezervorul. Pompa este pornită manual cu turaţia modificată după un grafic în timp şi generează debitul de fluid care este acumulat (integrat) de către rezervor.

Echipamentul de măsurare şi control este modelat de voltmetrul 30 şi osciloscopul cu două canale 40. Voltmetrul amplifică în prealabil semnalul măsurat. Pentru a avea indicaţii identice osciloscopul este conectat la ieşirea voltmetrului.

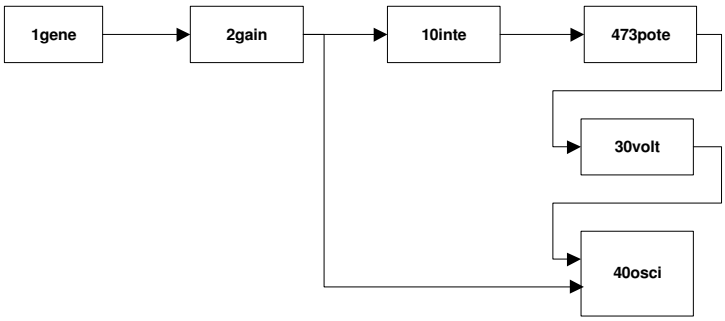


Fig. 10.2 Schema bloc a sistemului pompă-rezervor.

Studiul tehnic experimental al instalaţiei este dificil, costisitor şi durează destul de mult în cazul instalaţiilor lente, tocmai pentru că este realizat în timp real. Alte soluţii constau în folosirea unui model fizic sau matematic pentru instalaţia tehnologică şi sistemul de conducere automată. In ultimul caz calculatorul funcţionează în regim de simulare şi sistemul KitSAS permite realizarea programelor necesare. Kitul de Simulare şi Analiză a Sistemelor – KitSAS a fost realizat astfel încât folosirea sa să fie cât mai asemănătoare cu

acţiunile efectuate în cazul metodei experimentale. Pentru folosirea KitSAS se execută programul:

C:\KitSAS\Simlab32.exe

Apare panoul de comandă din *Fig. 10.3*. Acesta este format dintr-o fereastra, mai multe butoane şi un meniu. In fereastra panoului apar diferite mesaje pe care sistemul KitSAS le furnizează utilizatorului în legătură cu contextul prezent, erorile apărute sau mesaje feed-back la acţiunile efectuate de utilizator. De exemplu, la apariţia pentru prima oară a panoului de comandă apare un mesaj în care se specifică că nu există nici un sistem montat pe masa de lucru a KitSAS. Pentru a realiza o simulare trebuie să montăm sau să creăm un sistem acţionând butoanele corespunzătoare. Dacă sistemul există în magazie îl montăm acţionând butonul *Montare bloc, sistem*. Pentru exemplul considerat din Fig. 10.2 programul necesar se găseşte în fişierul:

C:\KitSAS\Magazie\Expo\SimpluGene.txt.

Fiecare element din figură este predefinit în KitSAS. Elementele au un mnemonic şi un număr, cuprins între 1 şi 999. Numărul este necesar deoarece este posibil ca acelaşi tip de element să apară într-un model de mai multe ori. În acest caz mnemonicul se păstrează dar numărul este diferit. Se observă în figură că pompa este modelată cu ajutorul elementelor cu numerele 1 şi 2. Elementul **gene** este un generator manual de semnal iar elementul **gain** este un amplificator care determină mărimea debitului pompei. Integratorul **inte** modelează rezervorul. Mărimea nivelului în rezervor este determinată de potenţiometrul **pote**.

După montare, în fereastra panoului de comandă KitSAS apare programul (*Fig. 10.3*) cuprins între cuvintele cheie *Program,* situat în prima linie şi *EndProgram*, din ultima linie. Intre aceste limite se găsesc comentarii şi descrierea blocurilor din Fig. 10.2. Acest program simplu a fost tehnoredactat într-o formă nu prea elegantă, dar care permite punerea în evidenţă a multor posibilităţi de prezentare. Comentariile pe mai multe linii sunt cuprinse între acolade. Comentariul pe o singură linie începe cu semnul //. Şirurile de caractere sunt prezentate între ghilimele. Între elementele programului se foloseşte drept separator spaţiul liber. Mnemonicele elementelor predefinite de kit sunt recunoscute drept cuvinte cheie. Se pot folosi litere şi cifre cu condiţia ca primul caracter să fie o literă. Caracterele mari şi mici sunt interpretate la fel. După mnemonice, între paranteze rotunde, se definesc parametrii fiecărui element. Dacă aceste definiţii lipsesc, parametrii respectivi iau valori implicite, de regulă valoarea zero pentru numere şi spaţiul liber pentru şirul de caractere. Într-o linie pot fi prezentate mai multe elemente sau descrierea unui element se poate întinde pe mai multe linii. Mnemonicul unui element şi numărul său pot fi alăturate sau separate printr-un spaţiu liber.

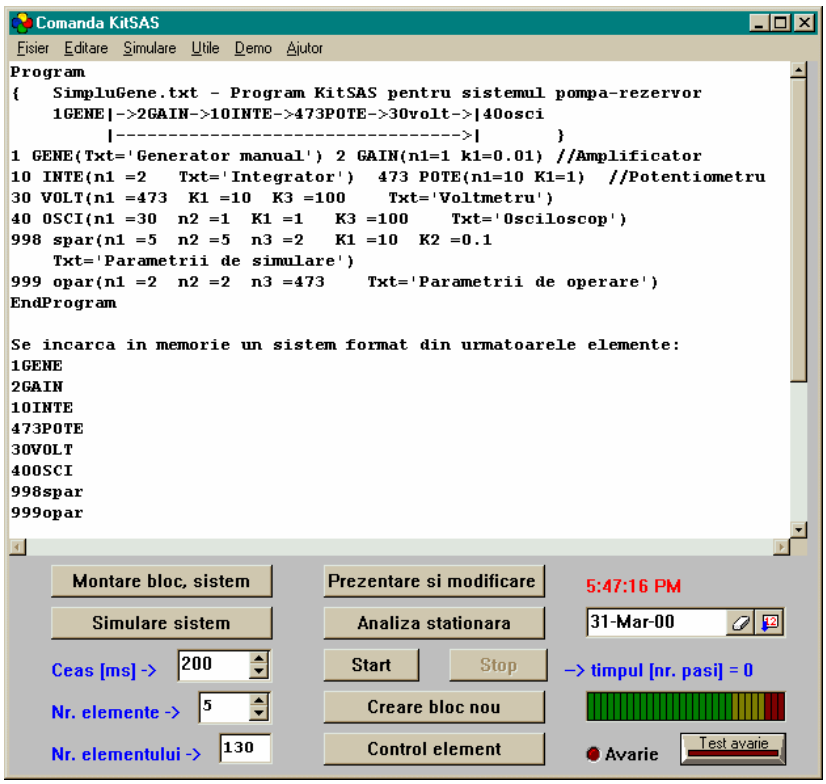


Fig. 10.3 Panoul de comandă KitSAS cu sistemul montat afişat în fereastra sa.

Fiecare element al kitului este un obiect caracterizat de opt parametrii:

- Trei numere întregi n1, n2 şi n3. De regulă aceste numere indică elementele la care se conectează intrările elementului caracterizate de aceste numere. KitSAS specifică întotdeauna unde sunt conectate intrările nu ieşirile.
- Trei numere reale K1, K2 şi K3. Acestea sunt valorile parametrilor elementului.
- Numărul real E care reprezintă valoarea iniţială a ieşirii.
- Şirul de caractere Txt. In cele mai multe cazuri acesta este o etichetă care cuprinde informaţii despre element.

Analizorul sintactic şi morfologic al kitului determină din program elementele sistemului şi parametrii lor. Dacă programul nu are erori se afişează în fereastra panoului de comandă numerele şi mnemonicele elementelor încărcate în memorie (*Fig. 10.3*).

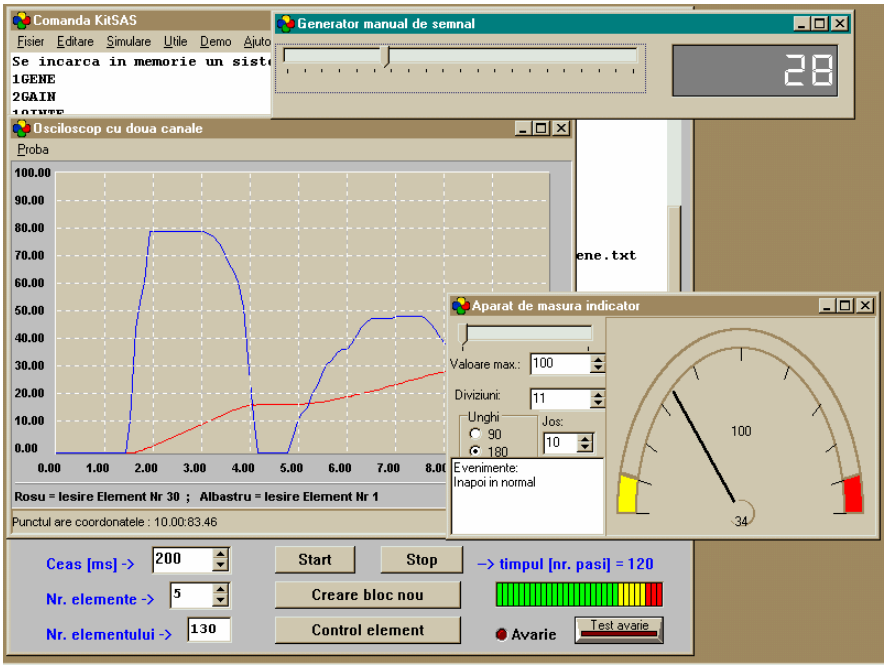


Fig. 10.4 Măsurări efectuate în timpul simulării sistemului pompă-rezervor.

Începerea simulării sistemului se face prin apăsarea tastei *Start.* La fiecare 200 ms se execută un pas de calcul. Dacă pasul de calcul este egal cu 200 ms atunci simularea se face în timp real. Simularea durează până când se apasă tasta *Stop.* La începerea simulării apar un generator manual de semnal, un osciloscop cu două canale şi un aparat de măsură indicator. Acestea sunt de fapt aparatele definite prin elementele programului *1gene, 30volt* şi *40osci.* Cu ajutorul generatorului se modifică debitul pompei şi se măsoară nivelul în rezervor cu cele două instrumente virtuale (Fig. 10.4).

După terminarea simulării apare un nou instrument virtual, înregistratorul numeric şi grafic cu două canale. Acesta este definit implicit de către kit şi apare la fiecare simulare prezentând rezultatele. Rezultatele numerice şi graficul pot fi salvate în diferite formate sau prelucrate cu ajutorul unui analizator de frecvenţă sau a programului Excel.

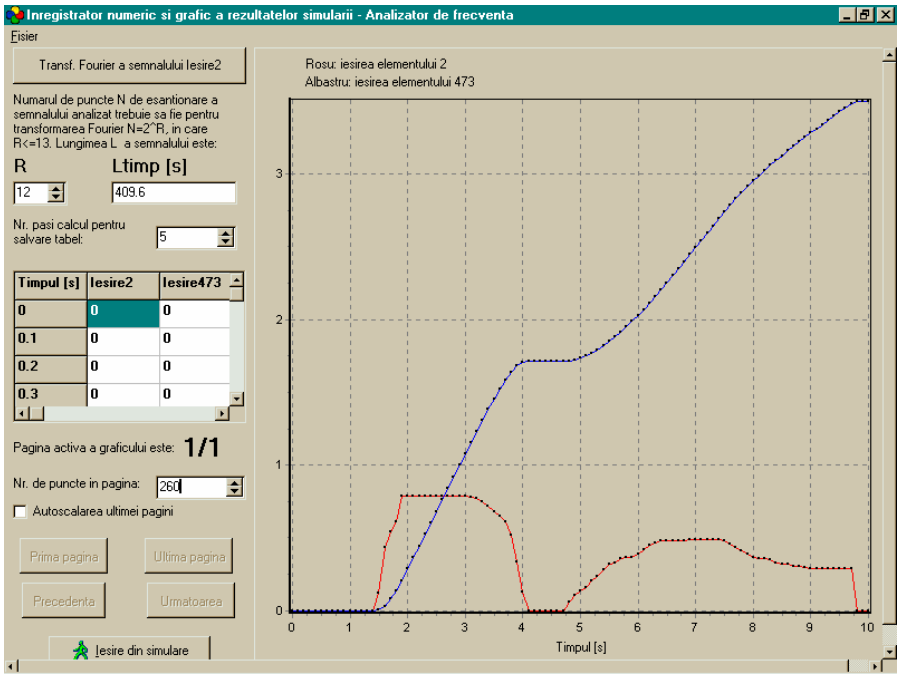


Fig. 10.5 Rezultatele înregistrate şi prelucrate după terminarea simulării.

Comanda *Prezentare şi modificare sistem*, *Fig. 10.3*, permite adăugarea şi ştergerea de elemente şi modificarea parametrilor fiecărui element. În felul acesta se pot construii noi sisteme pornind de la modele existente sau se poate simula un sistem cu diferiţi parametrii.

Dacă nu este necesară simularea în timp virtual se foloseşte comanda *Simulare sistem*, *Fig. 10.3*. În acest caz variabila timp nu este percepută şi măsurată în timpul simulării. Calculatorul execută simularea cât poate de repede şi instrumentele virtuale de măsură nu sunt folositoare. Rezultatele sunt obţinute cu ajutorul înregistratorului numeric şi grafic (Fig. 10.5) care întotdeauna apare la terminarea simulării. Înregistratorul are o memorie de 8192 valori măsurate. Reprezentarea grafică se face pe mai multe pagini şi se poate modifica numărul de puncte pe pagină.

KitSAS a fost proiectat astfel încât rezultatele simulării să poată fi analizate atât în domeniul timp cât şi în domeniul frecvenţă. După terminarea simulării rezultatele sunt prezentate întotdeauna în domeniul timp, Fig. 10.5. Scalarea pentru reprezentarea grafică pe două canale se face automat. Toţi paşii de calcul sunt înregistraţi numeric. Pentru tipărirea unui tabel cu rezultate este posibilă selectarea valorilor pe baza unui număr de paşi de calcul.

Dacă se doreşte transformarea în domeniul frecvenţă a primului semnal obţinut prin simulare (roşu) cu ajutorul Transformării Fourier Rapide se apasă tasta corespunzătoare (Fig. 10.5). Numărul punctelor de eşantionare a semnalului transformat trebuie să fie o putere a lui doi. Dacă numărul punctelor de eşantionare

ales este mai mare decât numărul valorilor obţinute prin simulare se atribuie valoarea zero punctelor suplimentare. Rezultatul transformării FFT este prezentat numeric şi grafic sub forma diagramelor Bode.

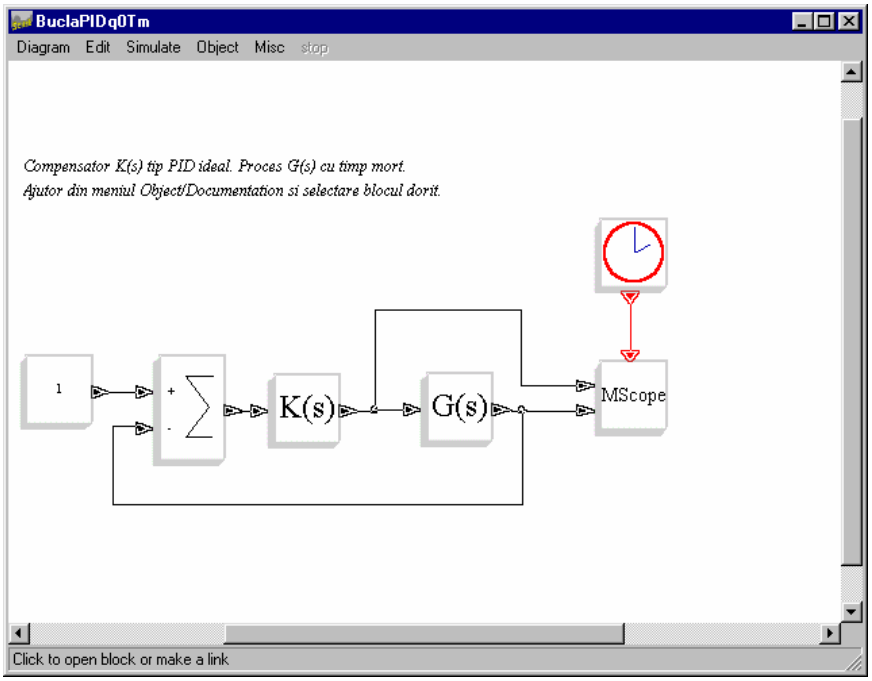


Fig. 10.6 Simularea unei bucle de reglare în Scilab/Scicos.

Pachetul de programe KitSAS permite analiza sistemelor electrice şi neelectrice nu numai prin simulare. De exemplu, circuitele electrice şi electronice pot fi analizate în regim sinusoidal staţionar, într-o manieră asemănătoare cu cea a programului Pspice, cu ajutorul comenzii *Analiza staţionară.*

KitSAS este un instrument puternic la îndemâna inginerilor, poate fi folosit la asistarea proiectării sau în scopuri educaţionale şi este o alegere care prezintă unele avantaje, în special de natură financiară, faţă de pachetele comerciale echivalente. Versiunea 6.11 este gratuită şi poate fi descărcată de la adresa www.geocities.com/larionescu.

10.2. Scilab / Scicos

Un mediu de dezvoltare a aplicaţiilor sistemice foarte cunoscut în mediul universitar şi în numeroase institute de proiectare este programul Matlab[204]. Există şi două programe gratuite pseudo-clone[205] cu Matlab: Scilab[206] şi Octave[207]

[204] Matlab împreună cu extensiile sale, cunoscute sub numele de toolboxes, are o documentaţie bogată şi numeroase exemple de aplicare. Din păcate este foarte scump. Informaţii despre Matlab se găsesc în numeroase locuri pe internet. Site-ul oficial este www.mathwork.com

În continuare se face o scurtă prezentare mediului de dezvoltare a aplicaţiilor sistemice Scilab[208].

Scilab este elaborat de INRIA – Institut National de Recherche en Informatique et Automatique. Are o documentaţie foarte bună în engleză, franceză şi spaniolă. La deschiderea mediului de dezvoltare Scilab apare o fereastră cu un prompter **-->** spre linia de comandă. În Fig. 10.7 se prezintă efectul executării a trei linii de comandă ce cuprind patru instrucţiuni.

Prima linie de comandă are două instrucţiuni, fiecare dintre ele terminată cu separatorul „punct şi virgulă". Linia de comandă se termină prin apăsarea tastei Enter. Prima instrucţiune atribuie variabilei a valoarea 2,5 iar a doua instrucţiune atribuie variabilei A valoarea complexă 2+3i. Se observă că denumirile variabilelor sunt sensibile la literele mari şi mici. Constanta imaginară %i este egală cu √-1.

A doua linie de comandă execută produsul variabilelor a şi A. Deoarece linia de comandă nu este terminată cu separatorul „punct şi virgulă" la terminarea comenzii prin apăsarea tastei Enter se va tipării rezultatul. De data aceasta se obţine numărul complex X=5+5i.

A treia linie de comandă execută un program, numit script, care se găseşte în fişierul text *alldems.dem* din directorul *SCI/demos*. SCI semnifică directorul în care a fost instalat Scilab, de obicei C://Program Files/Scilab-3.1.1. Rezultatul execuţiei scriptului *alldems.dem* constă în afişarea unei ferestre cu un meniu de alegere a unei demonstraţii de folosire a scilab. Prima opţiune este *Introduction to Scilab* care prezintă foarte bine posibilităţile mediului de dezvoltare a aplicaţiilor sistemic. Alte opţiuni de demonstrare se referă la *Graphics, Simulation, Control, Signal Procesing, Optimization, Graph & Networks, Scicos, Random, TK/TCL demos.* Scicos este programul de simulare asemănător cu Simulink din Matlab.

---

[205] Extensiile Scilab şi Octave nu mai sunt atât de asemănătoare cu vestitele toolboxes Matlab.

[206] Scilab poate fi descărcat gratuit împreună cu toată documentaţia şi exemplele de aplicare de la adresa: www.scilab.org

[207] Programul Octave, încă în curs de dezvoltare, poate fi descărcat gratuit de la adresa www.che.wisc.edu

[208] Prescurtare de la Scientific Laboratory şi din această cauză pronunţarea recomandată de autori este cea engleză şi nu franceză.

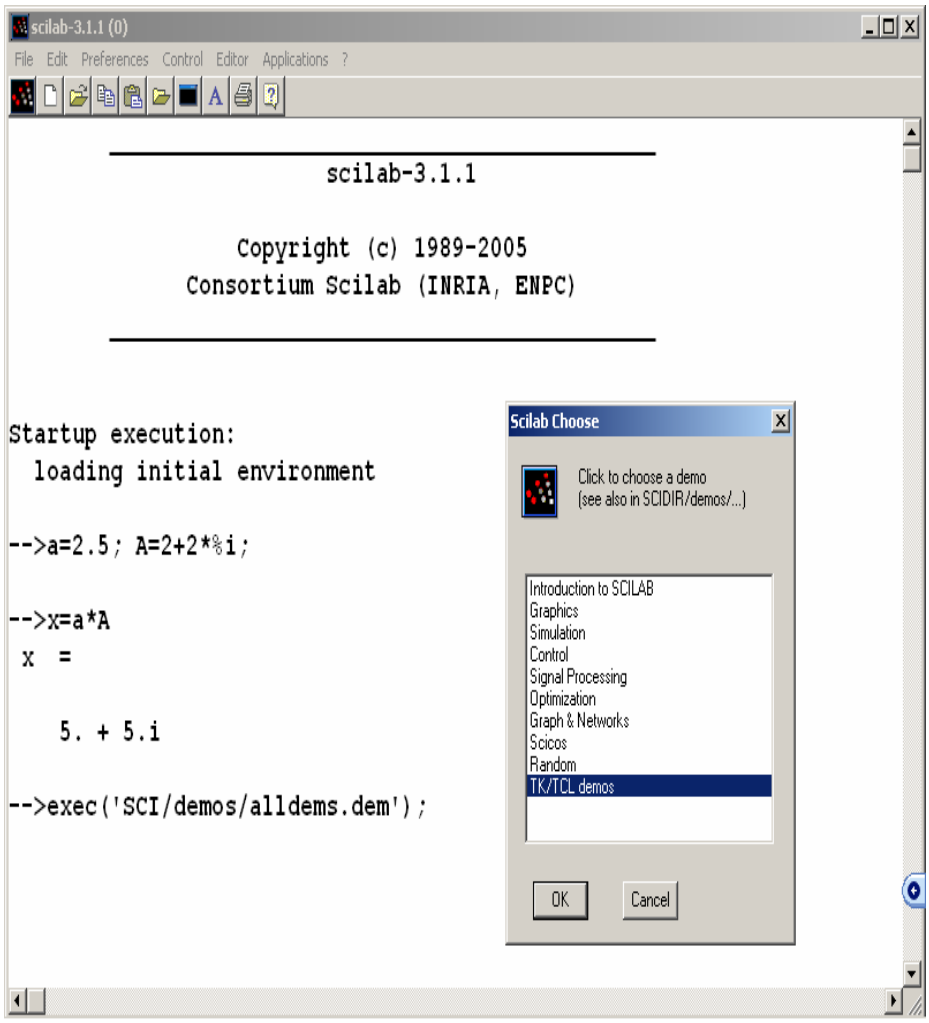


Fig. 10.7 Rezultatul executării în Scilab a patru instrucțiuni în trei linii de comandă

A treia linie de comandă poate fi generată şi din tabuul **?** al meniului care apare în fereastra Scilab din Fig. 10.7. Acest meniu conține şi opțiunea importantă *Scilab Help.* Din meniul *Aplication* al Scilab reținem opțiunea *Scicos* care afişează fereastra de editare grafică a simulatorului (Fig. 10.6) şi opțiunea *m2sci* care corespunde unui translator automat al programelor (scripturilor) Matlab în programe (scripturi) Scilab.

Din scurta prezentare făcută rezultă că programele Scilab pot fi introduse, la fel ca şi în Matlab, în două feluri: linie de comandă sau fişier text (script). Editorul de scripturi Scilab poate fi accesat din tabuul *Editor* al ferestrei Scilab.

Dacă avem un program Scilab sub formă de fişier text (script) atunci acesta poate fi executat cu ajutorul opțiunii *File/Exe…*din meniul ferestrei Scilab.

10.3. Scilab / ControlKit

Biblioteca de programe[209] ControlKit versiunea 3.0 pentru Scilab, elaborată de autor, reprezintă o colecţie de funcţiuni şi o interfaţă grafică cu utilizatorul destinate proiectării asistate de calculator a sistemelor de reglare automată. Programele se utilizează în stilul Matlab şi scilab, au un help incorporat şi valori implicite pentru fiecare comandă care permit acomodarea mai uşoară cu folosirea lor prin executarea rapidă a unui exemplu ghid de proiectare.

În prima etapă a proiectării se determină funcţia de transfer *nominală G(s)* a instalaţiei. Control Kitul are câteva scule care permit simularea identificarea în prezenţa zgomotului şi reducerea ordinului funcţiei de transfer.

A doua etapă a proiectării constă în determinarea unui model al lui G(s) folositor la stabilirea unei prime variante a compensatorului numită K1(s). Se foloseşte modelul din domeniul frecvenţă a lui Ziegler şi Nichols şi modelul din domeniul timp a lui Kupfmuller

O primă formă K1(s) a funcţiei de transfer a compensatorului este stabilită în etapa a treia de proiectare asistată de calculator. În acest scop se folosesc diferite reguli stabilite pe cale empirică folosind experimentări şi simulări. Compensatoarele obţinute în această etapă sunt diferite variante ale tipului PID.

Ultima etapă, a patra, a proiectării este dedicată modificării, dacă este cazul, a constantei de proporţionalitate a regulatorului stabilit în etapa precedentă astfel încât să fie îndeplinite diferite condiţii de stabilitate, performanţe şi robustete. În acest scop se foloseşte metoda locului rădăcinilor.

Diferite etape ale proiectării asistate de calculator sau părţi din etape pot fi omise pentru obţinerea unor rezultate aproximative.

10.3.1. Instalarea.

Se desarhivează fişierul care conţine ControlKit şi se copiază directorul obţinut în directorul lui scilab, de exemplu C:/Program Files/Scilab-3.1.1. În directorul ControlKit3.0 există două scripturi *builder.sce* şi *loader.sce.* Cu ajutorul comenzii *Exec…* din meniul *File* se execută:

exec('C:\Program Files\Scilab-3.1.1\ControlKit3.0\builder.sce');

În acest moment biblioteca este instalată în Scilab.

De fiecare dată când se doreşte utilizarea ei se execută în mod asemănător comanda:

exec('C:\Program Files\Scilab-3.1.1\ControlKit3.0\loader.sce');

10.3.2. Ajutor

După instalare se poate obţine ajutor selectând din meniu comanda:

?/Scilab Help/Graphics based tool for SISO system design

În directorul în care este instalat Scilab se va găsii după instalarea ControlKit şi fişierul QuickStart ControlKit. Se acordă atenţie versiunii Scilab şi ControlKit. Prezentarea din paragrafele următoare este valabilă pentru Scilab 3.1.1 şi ControlKit 3.0 şi, eventual, pentru versiunile următoare.

[209] Scripturi, adică fişiere text.

10.3.3. Pornirea şi utilizarea Control Kitului

Dacă s-a încărcat biblioteca ControlKit în Scilab apar următoarele mesaje:

Control Kit is loaded
enter "kit()" to start

În dreptul prompterului de comandă scilab se introduce comanda ***kit().*** Se afişează simultan ferestrele din Fig. 10.8 şi Fig. 10.9.

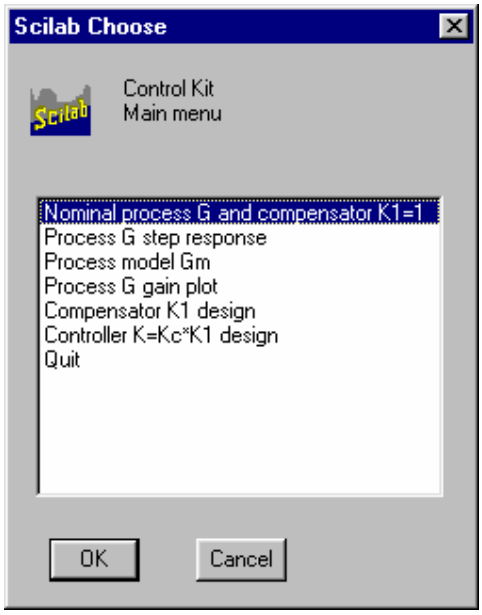


Fig. 10.8 Meniul principal al programului ControlKit

Se selectează succesiv valorile meniului din Fig. 10.8. Într-o primă abordare se pot accepta valorile propuse de kit. Singura dificultate poate apare la penultimul meniu în care se face proiectarea regulatorului K(s). Deoarece se foloseşte metoda grafică interactivă a locului rădăcinilor funcţiei de transfer este necesară selectarea cu mausul a unor puncte de pe grafic. Dacă nu se face această operaţie s-au este realizată defectuos se adoptă de asemenea o valoare implicită.

10.3.4. Proiectarea minimală cu ajutorul Control Kitului

Problema proiectării sistemului automat din Fig. 10.9 constă în determinarea compensatorului ***K(s)*** atunci când se cunoaşte instalaţia ***G(s)***

Etapele strict necesare din Fig. 10.8 pentru proiectarea asistată de calculator cu ControlKit sunt prima şi a cincia, adică introducerea funcţii de transfer nominale G(s) a procesului şi proiectarea compensatorului K1(s). Într-o primă aproximaţie regulatorul $K(s)=K_cK(s)$ din Fig. 10.9 este egal cu K1(s), adică se consideră constanta $K_c=1$.

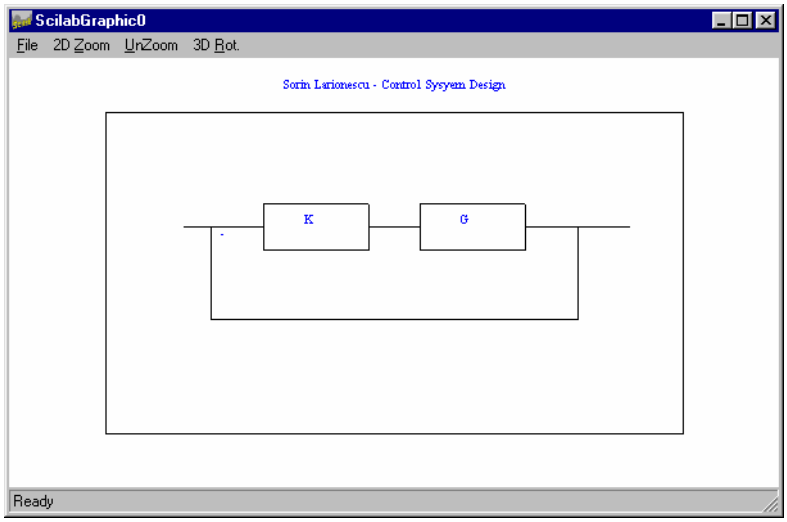


Fig. 10.9 Fereastra cu schema bloc care apare la începutul folosirii ControlKit

10.3.5. Proiectarea cu asigurarea stabilității şi robusteții

Schema bloc folosită de ControlKit este reamintită de fiecare dată când acesta este pornit prin afişarea ferestrei din Fig. 10.9. Din motive legate de metoda de proiectare folosită regulatorul şi procesul sunt despărțite în câte două blocuri aşa cum se vede în Fig. 10.10.

Funcția de transfer a procesului este descompusă într-o parte care reflectă efectul timpului mort şi într-o altă parte G1(s) care ia în considerare inerția şi amplificarea. Funcțiile de transfer ale procesului şi regulatorului apar la interfața de intrare cu utilizatorul în forma inginerească obişnuită, care explicitează constantele de proporționalitate şi constantele de timp. Folosirea eficientă a metodei de proiectare propusă de ControlKit, înțelegerea ei şi verificarea rezultatelor obținute necesită familiarizarea cu forma poli – zerouri a funcțiilor de transfer.

După introducerea[210] funcției de transfer G(s) a procesului calculatorul poate determina un compensator K1(s) de tip PID cu ajutorul comenzii a cincia din Fig. 10.8. În acest scop se folosesc două metode.

Prima metodă este cea elaborată de Ziegler şi Nichols şi constă în determinarea constantei de amplificare limită K_{limit} care conduce sistemul automat cu regulator de tip P şi proces G(s) la limita de stabilitate. Perioada oscilațiilor apărute este T_{limit}.

A doua metodă aproximează componenta G1(s) a procesului cu un model de tip Kupfmuller.

[210] Atenție la modul de introducere a fracțiilor raționale. Se urmăreşte exemplul implicit.

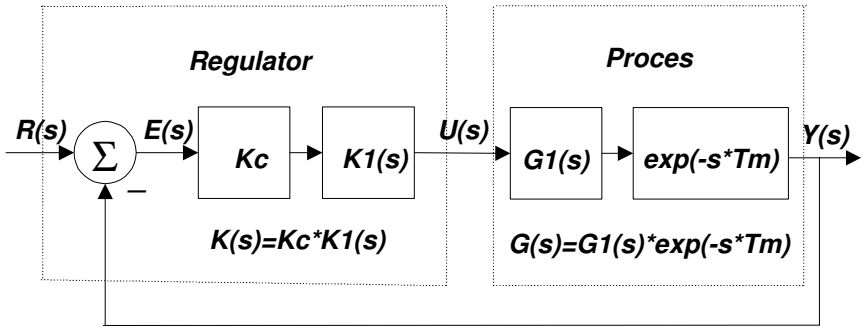


Fig. 10.10 Schema bloc pentru proiectarea asistată de calculator a sistemelor automate cu ajutorul ControlKit

După determinarea funcţiei de transfer K1(s) a compensatorului se trece la comanda a şasea din meniul principal Fig. 10.8 care foloseşte metoda locului rădăcinilor pentru calculului constantei K_c. Aceasta înmulţită cu K1(s) produce funcţia de transfer a unui regulator, prezentat în Fig. 10.10, care poate conduce automat procesul cu performanţele dorite.

În partea superioară a ferestrei care afişează locul rădăcinilor apare un meniu cu mai multe valori.

Meniul 2D Zoom permite focalizarea pe o regiune restrânsă a locului rădăcinilor. În felul acesta se poate selecta foarte exact punctul dorit de pe locul rădăcinilor. Comanda UnZoom reface forma iniţială a locului rădăcinilor.

Meniul Grid permite stabilirea domeniului performanţelor dorite pentru sistemul automat. O comandă foarte utilă din acest meniu este Zeta/Wn care stabileşte domeniul performanţelor impunând valori pentru fracţiunea de amortizare critică ζ şi pulsaţia naturală ω_n. Valoarea implicită a lui ζ este 0.22 şi corespunde acordării Ziegler – Nichols a regulatorului.

Meniul Response are multe valori care permit stabilirea performanţelor sistemului automat proiectat. O alegere frecventă din acest meniu este Step Response care furnizează performanţele standard.

Meniul Design realizează proiectarea asistată de calculator a regulatorului K(s). Valoarea cea mai importantă a meniului este "Compensator gain Kc" care determină constanta K_c. Cunoscând această valoare funcţia de transfer a regulatorului rezultă imediat din Fig. 10.10 deoarece K1(s) a fost determinat în etapa precedentă de proiectare.

Valorile "Phase Margin" şi "Gain Margin" ale meniului Design permit verificarea robusteţii stabilităţii şi performanţelor regulatorului proiectat.

În cadrul meniului Design există şi valoarea "Complementary Sensitivity T" care determină diagrama amplitudine – frecvenţă a funcţii de transfer a sistemului automat sub forma din Fig. 10.9. În mod frecvent această diagramă este folosită pentru determinarea benzii de trecere care este performanţa principală prin

care se apreciază capacitatea sistemului automat de a nu folosii mărimi de comandă U(s) prea mari şi care nu pot fi realizabile fizic.

Prima valoare a meniului Design este "Root Locus and gain Kloc". Această alegere din meniu este folosită dacă se modifică structura compensatorului K1(s) cu ajutorul comenzilor din meniul Edit.

Meniul Edit are multe valori cu care se pot introduce sau înlătura poli / zerouri ai compensatorului K1(s) sau se pot introduce / înlătura blocuri în cascadă cu K(s). Valoarea "Edit L=KG" permite modificarea funcţiei de transfer pe calea directă a buclei de reglare automată.

Meniul Settings are comenzi pentru stabilirea diferitor parametrii care intervin în proiectarea asistată de calculator. Valoarea "Root Locus" a acestui meniu permite modificarea constantei Kloc care are valoarea implicită 10. O altă alegere mai frecventă din acest meniu este "Dynamic Response" cu ajutorul căreia se stabilesc parametrii de simulare a sistemului automat prin comenzile din meniul Response.

O acţiune foarte importantă în această metodă de proiectare constă în selectarea cu mauseul a unui punct de pe locul rădăcinilor. Programul are grijă să ne amintească cât mai des de acest lucru.

După selectare se afişează valoarea constantei K_{loc}, polii p_i şi zerourile z_i care determină funcţia de transfer T(s) a sistemului automat.

10.4. Dynast

Programul Dynast este destinat *modelării*, *analizei* şi *simulării* sistemelor fizice dinamice multidisciplinare, sistemelor cu conducere automată şi altor tipuri de sisteme care pot fi modelate matematic. El poate fi executat on line pe internet la adresa *icosym.cvut.cz* sau off line folosind o versiune studenţească cu posibilităţi reduse[211] care poate fi descărcată gratuit de la aceeaşi adresă. Site-ul Dynast conţine o documentaţie extinsă, inclusiv multe exemple, referitoare la teoria sistemelor fizice multidisciplinare şi folosirea programului.

Programul Dynast furnizează sculele necesare pentru construirea a trei tipuri de modele sistemice[212]: *matematice*, *funcţionale* şi *structurale*. Dynast permite modelarea sistemelor folosind *simultan* toate cele trei tipuri de modele. Toate modelele sunt descrise cu ajutorul unor fişiere text care pot fi construite cu ajutorul unui editor incorporat.

Modelele matematice pot fi de forma unor sisteme de ecuaţii *neliniare* algebrice sau diferenţiale *explicite* sau *implicite*.

Modelele funcţionale sunt de tipul unor scheme bloc. Elaborarea fişierelor text care descriu modelele funcţionale se poate face şi cu ajutorul unui editor grafic de scheme incorporat.

Modelele structurale sunt de tipul unor reţele electrice, fluidice, termice sau mecanice. Ca şi în cazul modelelor funcţionale sub formă de scheme bloc, modelele structurale sub formă de reţele pot fi elaborate în afara de editorul de

[211] Versiunea off line studenţească poate rezolva maximum opt ecuaţii diferenţiale neliniare. Versiunea on line nu are limitări.

[212] Modele sistemice numite sisteme de către program.

texte incorporat şi cu ajutorul editorului grafic de scheme conţinut printre sculele programului Dynast.

Analiza modelelor sistemice cu ajutorul programului Dynast poate fi de următoarele tipuri:

1. Analiza neliniară în regim tranzitoriu.
2. Analiza în regim staţionar pentru semnale de intrare sinusoidale[213].
3. Analiza semisimbolică pe baza transformatei Laplace.
4. Analiza în domeniul timp.
5. Analiza Fourier în domeniul frecvenţă[214].

Simularea sistemelor este considerată de către Dynast drept o analiză în domeniul timp sau frecvenţă.

10.4.1. Modelul matematic neliniar explicit

Într-o primă aplicaţie a mediului de dezvoltare Dynast se simulează comportarea statică a robinetului de reglare pe baza unui model matematic sub forma unei ecuaţii algebrice neliniare explicite.

$$q = \frac{1}{\sqrt{1+\psi(\frac{1}{k_v^2}-1)}} \qquad (10.1)$$

Variabila q reprezintă debitul normalizat reglat de robinet. Autoritatea de reglare ψ a robinetului este egală cu raportul dintre căderea de presiune pe robinet cu robinetul deschis şi căderea de presiune pe robinet cu robinetul închis (căderea de presiune pe sistem).

Constanta k_v pentru robinete liniare este:

$$k_v = k_{v0} + (1-k_{v0})h \qquad (10.2)$$

Constanta k_v pentru robinete logaritmice este:

$$k_v = e^{n(h-1)} \qquad (10.3)$$

în care h este deplasarea normalizată a tijei robinetului cu valori cuprinse între 0 şi 1.

Pentru această aplicaţie se consideră următoarele valori pentru constantele k_{v0} şi n:

$$k_{v0} = 0.02$$
$$n = \ln\frac{1}{k_{v0}} \qquad (10.4)$$

[213] Această analiză este numită analiză în curent alternativ în cazul modelelor electrice sau electronice sau analiza la vibraţii în cazul sistemelor mecanice.

[214] Determinarea diagramelor Bode.

Autoritatea de reglare ψ are valoarea:

$$\psi = 0.1 \tag{10.5}$$

10.4.2. Instrucţiuni pentru elaborarea programelor DYNAST

Modelul matematic explicit neliniar elaborat în paragraful precedent să-l introducem într-un program Dynast. Etapele sunt următoarele:

1. Se deschide interfaţa (DYNAST Shell) a mediului de dezvoltare a aplicaţiilor DYNAST.
2. Se alege eventual meniul *Help* pentru a putea consulta documentaţia: User's Guide, Tutorials, Examples, Modeling Basics, Course on Modeling, Glossary.
3. Pentru elaborarea unui program nou se alege meniul *File/New* pentru a începe o nouă problemă şi se selectează opţiunile implicite din fereastra care apare: *File Type = Problem text* şi *Template = Standard template*. Se completează numele fişierului în care se va păstra programul (LarionescuRobinet) şi numele programului care va apare la începutul lui (S Larionescu – Caracteristica statica a robinetului de reglare).
4. Intr-o fereastră separată apare următorul şablon (template) al programului:

```
*: S Larionescu – Caracteristica statica a robinetului de reglare
*SYSTEM;

*END;
```

Între cuvintele *SYSTEM* şi *END* se defineşte modelul sistemului, în această aplicaţie un model matematic de tip ecuaţie explicită. Ecuaţia se scrie între aceste cuvinte în mod obişnuit, respectând simbolurile folosite curent în programare pentru operaţii şi funcţii care sunt prezentat în meniul de *Help*.
Un exemplu de program cu ecuaţii explicite simple este următorul:

```
*: S Larionescu - Caracteristica statica a robinetului de reglare
*SYSTEM;
n=log(50);
h=0.5;
psi=0.15;
y=1/(sqrt(1+psi*h));
*TR;
DC h 0 1;
PRINT(501) y;
RUN;
*END;
```

Se observă că fiecare instrucţiune se termină cu separatorul punct şi virgulă.

5. Se defineşte tipul de analiză care se realizează cu ajutorul modelului. În acest scop se selectează meniul *Analysis/Numerical Nonliniar/Analysis*. In fereastra care apare se alege tipul de analiză parametric cu parametrul h care variază între 0 şi 1. Ieşirea analizei *Output* se calculează în 501 puncte (*Equidistant output*). Se selectează meniul *Analysis/Nonliniar analysis/Desired variables* şi

se alege pentru exemplul de mai sus variabila de ieşire y. Toate opţiunile de analiză sunt traduse imediat în instrucţíuni de program.

6. Se alege meniul *Run/Analysis & Plot*. In fereastra care apare se prezintă graficul lui y în funcţie de h.
7. Meniurile *Axis* şi *Plot* permit câteva alegeri interesante a modului de prezentare a graficului. De exemplu *Plot/Tracing* afişează coordonatele numerice ale poziţiei cursorului pe curba afişată. Ieşirea din acest regim se face apăsând tasta *Esc*. Unele opţiuni se pot alege şi cu click dreapta.

O variantă mai elaborată a programului este următoarea

```
*: Caracteristica de lucru statica a robinetului de reglare
*SYSTEM;
h=0.5; :parametru
kv0=0.05;
n=log(20);
kvlin=kv0+(1-kv0)*h; :varianta liniara
kvlog=exp(n*(h-1)); :varianta logaritmica
psi=0.1; :autoritatea de reglare
qlin=1/(sqrt(1+psi*(1/(kvlin*kvlin)-1)));
qlog=1/(sqrt(1+psi*(1/(kvlog*kvlog)-1)));

*TR;
DC h 0 1;
PRINT(501) kvlin, kvlog, qlin, qlog;
RUN;

*END;
```

10.4.3. Modelul matematic neliniar implicit

Se simulează comportarea dinamică a sistemului pompă - rezervor din Fig. 3.2 pe baza unui model matematic implicit sub forma unei ecuaţii diferenţiale neliniare. Aplicând legea conservării masei[215] la sistemul pompă + rezervor din Fig. 3.2 rezultă următoarea ecuaţie diferenţială neliniară:

$$\frac{d}{dt}(A\rho h_2) = w_1 - w_3 = w_1 - \alpha\sqrt{h_2} \qquad (10.6)$$

[215] Masa acumulată în rezervor este egală cu diferenţa dintre debitul masic de intrare şi debitul masic de ieşire.

sau

$$A\rho\frac{dh_2}{dt}+\alpha\sqrt{h_2}=w_1 \qquad (10.7)$$

Considerând variații mici ale nivelului Δh_2 în jurul nivelului staționar h_0 se obține:

$$A\rho\Delta(\frac{dh_2}{dt})+\alpha\Delta(\sqrt{h_2})=\Delta w_1 \qquad (10.8)$$

Pentru variații foarte mici operatorul Δ poate fi înlocuit cu operatorul diferențial d.

$$d(\sqrt{h_2})=\left.\frac{d\sqrt{h_2}}{dh_2}\right|_{h_2=h_0} dh_2=\frac{1}{2\sqrt{h_0}}dh_2 \qquad (10.9)$$

Pentru simplificarea scrierii notăm $dh_2 = h$ şi $dw_1 = w$. Ecuația diferențială, de data această liniară, este:

$$A\rho\frac{dh}{dt}+\frac{\alpha}{2\sqrt{h_0}}h=w \qquad (10.10)$$

Pentru sistemele liniare se poate definii capacitatea C şi rezistența R:

$$C\frac{dh}{dt}+\frac{1}{R}h=w \qquad (10.11)$$

Această ecuație este valabilă numai pentru variații mici ale nivelului h în jurul valorii h_0.

Pentru aplicația numerică se consideră

$$A\rho=1,\quad \alpha=2 \qquad (10.12)$$

Rezultă:

$$C=1,\quad R=6 \qquad (10.13)$$

Pentru simulare considerăm că în starea inițială nivelul în rezervor este h2=36. Aceasta este condiția inițială pentru simularea cu model neliniar. Asigurarea acestui nivel în regim staționar, adică dh2=0, necesită un debit w1=12, care rezultă din (10.7). Dacă debitul w1 creşte brusc[216] cu 1,2 adică ajunge la

[216] Sub forma unei funcții treaptă.

valoarea w1=13,2 atunci şi nivelul h2 se modifică în timp. Simularea sistemului cu modelul neliniar (10.7) ne arată variaţia lui h2.

Modelul neliniar poate fi liniarizat rezultând relaţia (10.11). Deoarece liniarizarea este valabilă doar jurul punctului de liniarizare corespunzător h2=36 şi w1=12, de data aceasta nivelul iniţial este h=0 iar w=1,2. Pentru a se putea compara simularea în cele două situaţii, cu model neliniar şi cu model liniar, se introduce pentru modelul neliniar variabila y=h2-36 care arată variaţia nivelului în jurul punctului de liniarizare. Programul Dynast este prezentat în continuare.

```
*: Model neliniar si liniarizat al sistemului pompa rezervor
: A2PompaRezervorMic.prb
: Semnalul de intrare mare pentru modelul neliniar este debitul w1=12
: Semnalul de intrare mic pentru modelul liniar este debitul w=1.2
: Semnalul de ieşire este nivelul h2 mare pentru modelul neliniar
: sau h pentru modelul liniar
*SYSTEM;
SYSVAR h2, h;
: Sistemul1, neliniar pentru semnal mare w1=13.2
: Nivelul initial este h2=36
0 = VD.h2+2*sqrt(h2)-13.2;
: Sistemul2, liniar pentru semnal mic w=1.2
: Nivelul initial este h=0
0 = VD.h+(2/(2*sqrt(36)))*h-1.2;
: y este variatia mica a nivelului h2 peste valoarea initiala 36
y=h2-36;
*TR;
TR 0 50;
PRINT(501) y, h;
INIT h2=36, h=0;
RUN;
*END;
```

Dacă se execută[217] programul se constată că pentru semnale mici modelul liniar dă aproximativ aceleaşi rezultate ca şi modelul neliniar în jurul punctului static de funcţionare.

10.4.4. Modelul structural sub forma de reţea

Se simulează comportarea dinamică a sistemului pompă - rezervor din Fig. 3.2. Pentru aceasta din meniul programului Dynast se alege comanda:

File/New/Diagram

În fereastra care apare se desenează modelul structural sub formă de reţea a sistemului pompă – rezervor. Se obţine reţeaua din Fig. 10.11. Pentru desenare se alege din meniul programului Dynast comanda:

Place/Diagram Part/Pure physical element/fluid or acustic domain

[217] Simularea poate fi făcută off line sau on line pe internet. Trecerea de la un regim la altul se face selectând meniul *Preferences/Options/Solver.*

Din fereastra care apare se trag simbolurile dorite. Click stânga pentru plasarea simbolului, click dreapta pentru terminarea acțiunii. Un simbol selectat poate fi rotit spre dreapta prin apăsarea tastei R, poate fi oglindit pe verticală prin apăsarea tastei Y sau poate fi oglindit pe orizontală prin apăsarea tastei X. După selectarea cu click dreapta a elementelor sistemului se poate edita numele şi valoare sa.

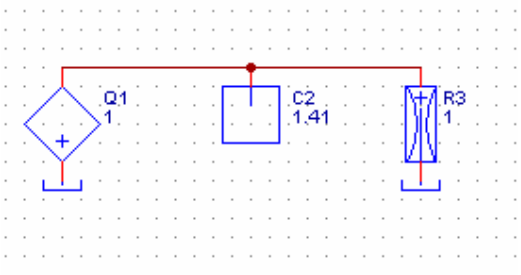


Fig. 10.11 Modelul structural al sistemului pompă rezervor din Fig. 3.2

După terminarea desenării se alege din meniu comanda:

View/Problem Text

Apare următorul program care descrie structura şi valorile elementelor sistemului structural sub formă de rețea.

```
*: Retea pompa rezervor
*SYSTEM;
C2 1 = 1.41;
Q1 > J 0-1 = 1;
R3 > G 1 = 1;
*END;
```

Programul nu spune însă nimic despre tipul de analiză pe care dorim să îl realizăm. Pentru aceasta alegem din meniu comanda:

Analysis/Numerical Nonliniar

Apare fereastra din Fig. 10.12 şi se aleg parametrii analizei, variabilele dorite şi condițiile inițiale.

La apăsarea tastei OK programul este completat automat de către DYNAST şi apare sub forma următoare:

```
*: Retea pompa rezervor
*SYSTEM;
C2 1 = 1.41;
Q1 > J 0-1 = 1;
R3 > G 1 = 1;
*TR;
TR 0 10;
PRINT(501) V.C2;
INIT V.C2=0, V.1=0;
RUN;
*END;
```

Se simulează sistemul cu ajutorul comenzii:

Run/Anaysis & Plot

Se personalizează graficul cu ajutorul a numeroase opţiuni şi se tipăreşte.

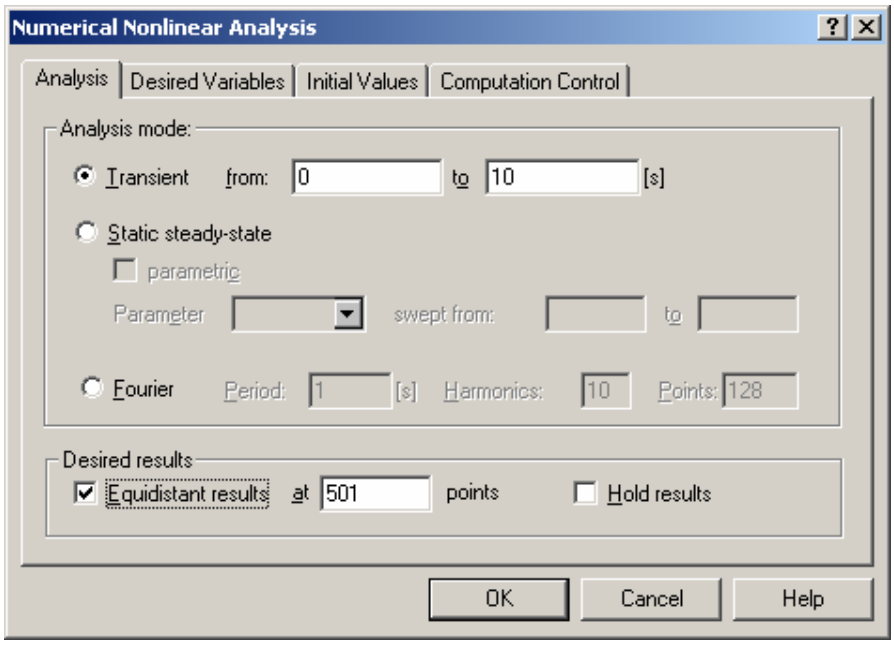


Fig. 10.12 Fereastra pentru alegerea parametrilor analizei.

10.4.5. Modelul structural neliniar

Se simulează comportarea dinamică a sistemului pompă - rezervor din cursul C3, figura 1, pe baza unui modelului structural din figura 2, acelaşi curs. Spre deosebire de aplicaţia 3, de data aceasta circuitul hidraulic este considerat în două variante: cu rezistenţa hidraulică neliniară şi cu rezistenţa hidraulică liniarizată. Comparând relaţiile (4) şi (5) rezultă rezistenţa şi conductanţa hidraulică:

$$w_3 = \frac{\alpha}{\sqrt{h_3}} h_3 = G h_3$$

$$G = \frac{\alpha}{\sqrt{h_3}} \qquad (10.14)$$

$$R = \frac{\sqrt{h_3}}{\alpha}$$

Se observă că conductanţa hidraulică nu este constantă ci depinde neliniar de mărimea transversală pe rezistenţă.

Liniarizarea rezistenţei în jurul nivelului h_0 se face astfel:

$$
\begin{aligned}
h_3 &= Rw_3 \\
h_3 &= \frac{\sqrt{h_3}}{\alpha} w_1 \\
\sqrt{h_3} &= \frac{1}{\alpha} w_1 \\
d(\sqrt{h_3}) &= \frac{1}{2\sqrt{h_0}} dh_3 = \frac{1}{\alpha} dw_1 \\
h &= \frac{2\sqrt{h_0}}{\alpha} w
\end{aligned}
\qquad (10.15)
$$

Programul Dynast este următorul:

```
*:Comparare circuit neliniar liniar
*SYSTEM;
: Circuit neliniar
: Starea (nivelul) initiala V.C1=0
C1 > C 1-0 = 1;
J1 > J 0-1 = 12;
G1 > G 1-0 = 2/sqrt(V.G1);
: Circuit liniar
: Starea (nivelul) initiala V.C2=0
C2 > C 2-0 = 1;
J2 > J 0-2 = 12;
G2 > G 2-0 = 2/(2*sqrt(36));
*TR;
TR 0 50;
PRINT V.C2, V.C1;
: Se observa constante de timp diferite
INIT V.C1=0, V.C2=0;
RUN;
*END;
```

10.4.6. Modelul funcțional.

Se simulează o instalație pentru o intrare treaptă şi cu următoarea funcție de transfer:

$$
G(s) = K_f \frac{1}{1 + 0.1T_f s + T_f^2 s^2} \qquad (10.16)
$$

$K_f = 1$

$T_f = 1$

Un exemplu tip de program de simulare este următorul:

```
*:ModelFunctionalSimplu
```

```
: Raspunsul unui bloc cu functia de transfer 1/(1+0.1s+s^2) la o
: intrare rampa

*SYSTEM;                  :definirea sistemului functional
BSgen U=time*(time>0);    :generator de impuls rampa
num /poly/ 1;             :polinomul de la numarator
den /poly/ 1, 0.1, 1;     :polinomul de la numitor
BT1 Y=num/den*U ;         :definirea blocului cu o intrare U,
                          :o iesire Y si o functie de transfer

*TR;                      :tipul si parametrii simularii
TR 0 20;                  :intervalul de simulare
PRINT(111) Y, U;          :se tipareste intrarea U si iesirea Y
RUN;                      :in 11 puncte

*END;
```

10.5. Simulatorul APL Trilogi

În continuare se prezintă un exemplu de folosire a automatelor programabile logice Trilogi Tl4 sau Tl5 pentru implementarea automatului cu evenimente discrete pentru comanda unei benzi rulante dintr-o hală de producţie.

Sistemele software Trilogi Tl4 pentru DOS şi Tl5 pentru Windows au mai multe funcţiuni. In continuare vom numii aceste sisteme simulatoare deşi poate nu aceasta este funcţiunea lor cea mai importanta. Simulatoarele pot fi descărcate gratuit de la adresa www.tri-plc.com/trilogi.htm.

10.5.1. Simulatorul Trilogi Tl4 pentru DOS

Pornirea simulatorului sub Windows. Se executa programul simulatorului ***TI4eval exe*** din catalogul Tl4.

Pornirea simulatorului sub DOS. La apariţia prompterului DOS se executa programul ***DC*** care este navigatorul numit Dos Commander. Cu ajutorul lui se selectează si se executa programul simulatorului ***TI4eval.exe.***

Simulatorul funcţionează in trei moduri de operare diferite:

a) simulare/configurare/compilare,

b) navigare,

c) editare.

In modul de operare ***simulare/configurare/compilare*** este activa bara superioara de comenzi, numita *Meniu* si se realizează următoarele funcţiuni:

1. Simulează in timp real pe calculatoare PC sub sistemul de operare DOS funcţionarea anumitor tipuri de automate programabile.
2. Configurează automatul programabil, gestionează comunicarea dintre PC si automatul programabil si compilează programele tip schema desfăşurata întocmite pentru automate programabile in instrucţiuni specifice si le transfera acestora.
3. Compilează schema desfăşurata in instrucţiuni specifice automatului programabil.

4. Gestionează comunicarea dintre Calculatorul PC si automatul programabil

In modul de operare ***navigare*** (Browse) se poate deplasa cu ajutorul tastelor săgeţi printre circuitele schemei desfăşurate care reprezintă programul automatului programabil simulat. Acum este activa bara de comenzi inferioara.

In modul de lucru ***editare*** se modifica sau se creează circuitele *individuale* ale schemei desfăşurate selectate prin navigare. Din aceasta cauza se poate trece in modul editare numai după ce am intrat in modul navigare si am selectat circuitul dorit. Comenzile se dau tot cu ajutorul barei inferioare care are de data aceasta alt conţinut.

La pornirea simulatorului de automat programabil (AP) trebuie sa se încarce (File/*Load)* un program existent sau sa se creeze un program nou *(File/New)!*

Simulatorul are doua bare cu comenzi. Bara superioara este numita *Menu* si conţine comenzile simulatorului, configuratorului si compilatorului iar bara inferioara este numita *Browse* si conţine comenzile navigatorului sau editorului. Activarea unei bare sau alta se face cu tasta ***ESC***. Aceasta tasta este folosita foarte frecvent si pentru a scăpa de unele comenzi pe care le solicita programul. Ajutor pentru fiecare comanda din cele doua bare se obţine cu tasta *F1*.

Pentru navigarea printre comenzile simulatorului se recomanda folosirea tastelor cu săgeţi, ESC, TAB si nu a mauselui.

Oprire. Se foloseşte comanda *File/Quit* sau se apasă simultan tastele *Alt-X*. Pentru aceasta este necesar ca bara superioara de comenzi (Menu) sa fie activa.

10.5.2. Încărcarea unui program existent.

Se foloseşte comanda *File/Load* din bara superioara de comenzi. Simulatorul solicita numele programului, Daca se foloseşte astericul (wildcard) in numele programului, de exemplu *.pc4, se deschide o fereastra si se poate naviga in directorul curent pentru alegerea programului dorit. Un exemplu este: *autpor.pc4* sau *demo.pc4.*

10.5.3. Elaborarea unei scheme desfăşurate noi.

Se foloseşte comanda *File/New*. cu care se trece in modul de lucru navigare in care este activa bara inferioara de comenzi. Doua comenzi sunt mai importante. Cu Tasta Ins se activează meniul *Circuit* iar cu tasta *spbar* se activează modul de lucru *editare* care are o bara inferioara proprie de comenzi. Aceste comenzi pot adaugă circuite noi la schema desfăşurata. Cu tasta ESC se poate reveni in celelalte moduri de lucru: simulare/configurare sau navigare.

10.5.4. Simularea funcţionarii unui automat programabil.

Se încarcă sau se elaborează schema desfăşurata care constituie programul automatului programabil. Se trece, daca este necesar, ca ajutorul tastei ESC, in regimul simulare/configurare/compilare in care este activa bara superioara Menu de comenzi. Se executa comanda Simulate/Continue run sau se apasă tasta F9. Apare un panou in care sunt vizibile toate contactele si bobinele automatului programabil. Cu ajutorul săgeţilor se pot selecta intrările. Valoarea fiecărei intrări (contact) poate fi basculata (trecuta din starea “acţionat” in starea “neacţionat” şi

invers) cu comanda in nivel[218] ***Enter*** sau comanda in impuls[219] ***Ctrl-Enter***. Starea intrărilor si bobinelor este indicata de prezenta unui LED. Daca in aceasta situaţie se trece cu ajutorul tastei ESC in modul de lucru ***navigare*** se observa starea contactelor si bobinelor, acţionate sau neacţionate.

10.5.5. Configurarea automatului programabil.

Înainte de elaborarea unui nou program sub forma de schema desfăşurata trebuie sa se configureze automatul programabil, adică sa se dea denumiri intrărilor si bobinelor. Aceasta se face in modul de lucru simulare/configurare/compilare cu comenzile Edit/Input, Edit/Output, Edit/Relay, etc.

10.5.6. Introducerea unui comentariu.

Se trece in modul navigare (cu tasta Esc) in care este activa bara inferioara de comenzi si se executa comanda Ins/Put coments.

10.5.7. Editarea unei scheme desfăşurate.

Pentru a folosi acest mod de lucru este necesar sa existe o schema desfăşurata obţinuta prin comenzile File/Load sau File/New ale barei superioare. Când se creează un program nou schema este goala, nu are nici un circuit. Se selectează circuitul care urmează sa fie editat. Daca schema este goala acesta este in mod implicit circuitul #1. Se trece in modul de operare editare cu comanda Spbar. Cursorul începe sa pâlpâie. Se deplasează cursorul cu ajutorul tastelor săgeţi selectând elementele existente de circuit daca acestea exista. Odată elementul de circuit selectat se selectează cu ajutorul tastei Tab poziţia in care se doreşte introducerea unui element de circuit, contact sau bobina, de exemplu. Se apăsa tasta Ins. Apare un meniu din care se poate selecta tipul elementului care sa fie introdus. Semnificaţia fiecărui simbol se obţine solicitând ajutor cu tasta F1. Pe scurt sunt elemente de circuit (contacte normal deschise, contacte normal închise sau bobine) care pot fi conectate in serie, in paralel sau in paralel/serie. Conectarea serie sau paralel nu pune probleme. Eventual câteva încercări ne familiarizează cu procedura. Mai dificila este conectarea paralel serie. Câteva îndrumări pentru realizarea unei astfel de conexiuni. Se determina punctul din care trebuie sa plece ramura paralela. Pentru aceasta se selectează mai întâi cu ajutorul tastelor săgeţi un element si apoi cu tasta Tab punctul din stânga sau dreapta a acestui element. Daca exista mai multe ramuri paralele in circuitul editat se selectează cu tastele săgeţi si tasta Tab punctul de *conexiune* a ramurilor dorit care este diferit *borna* unui element. Se apăsa tasta Ins pentru apariţia meniului Ins Element. Din acest meniu se selectează elementul cu numărul 5,6 sau 10. Cursorul se transforma intr-un x indicând punctul de pornire al ramurii paralele. Cu tastele săgeţi se deplasează cursorul in poziţia in care se doreşte sa fie punctul terminal al ramurii paralele. Din tabela de configurare dorita (de obicei tabela Input) se alege denumirea elementului (contactului). Daca nu apare tabela dorita (de exemplu Input) cu tastele săgeţi se navighează pana când apare aceasta. Se apasă Enter si elementul

[218] Comanda se memorează.
[219] Comanda nu se memorează.

ales este conectat in paralel intre punctele dorite. Daca dorim ca in *serie* cu acest element sa introducem un altul se selectează elementul, apoi unul din capetele acestuia si in final se apasă tasta Ins care declanşează secvenţa de operaţii necesara.

10.5.8. Prezentarea tabelului de configurare si a schemei desfăşurate.

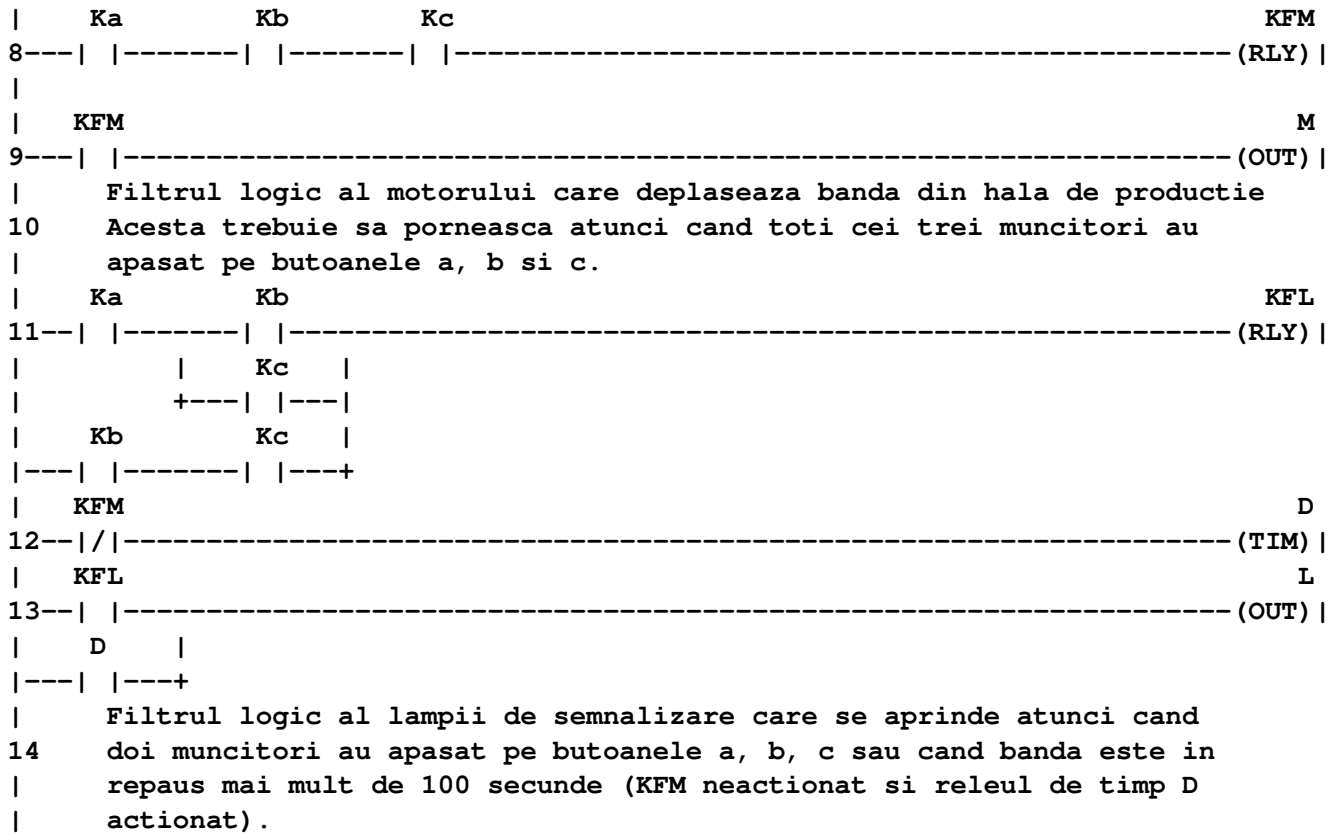
In modul de operare simulare/configurare/compilare se executa comenzile Print/I/O Tables si Print/Label-name Logic. Se poate alege tipărirea intr-un fişier sau la imprimanta. Fişierele rezultante sunt de tip text.

10.5.9. Un exemplu de program

In continuare se prezintă tabela de configurare si programul AP sub forma de schema desfăşurata pentru automatul cu evenimente discrete care conduce o banda rulanta dintr-o hala de producţie. Pentru simulare se observa ca toate intrările sunt de tip impuls si acţionarea lor se face prin apăsarea tastelor Ctrl – Enter. La apăsarea in impuls a doua dintre butoanele a, b sau c trebuie sa se aprindă lampa de semnalizare L avertizând pe al treilea muncitor ca operaţiile celorlalţi doi muncitori sunt terminate si trebuie sa se grăbească. La apăsarea in impuls, de către muncitori, a tuturor celor trei butoane trebuie sa pornească motorul M care acţionează banda rulanta. Când banda a avansat intr-o noua poziţie un palpator acţionează in impuls contactul traductorului t. Sistemul revine in situaţia iniţiala cu lampa stinsa si motorul oprit. Atunci când banda rulanta este in repaus mai mult de 100 secunde (motorul nu funcţionează) se aprinde lampa L care semnalizează ca nu se lucrează in ritmul normal. Daca se apasă tasta Esc in timpul simulării se poate observa pe schema desfăşurata starea contactelor si a bobinelor.

```
------------------------------------------------------------------------
                        Combined I/O Definition Table
    ------------------------------------------------------------------------
     I/O#|  Input   |  Output  |  Relay   | Timer    Value | Counter   Value
    ------------------------------------------------------------------------
       1 | a        | L        |          | D        1000  |
       2 | b        | M        |          |                |
       3 | c        |          | Ka       |                |
       4 | t        |          | Kb       |                |
       5 |          |          | Kc       |                |
       6 |          |          | KFL      |                |
       7 |          |          | KFM      |                |
       8 |          |          |          |                |
    ------------------------------------------------------------------------
```

```
                                                                BANDA.PC4
                                                                =========
|     Automatul cu evenimente discrete pentru comanda unei benzi rulante
1     dintr-o hala de productie.
|     ************************************************************
|     Trei muncitori apasa butoanele in impuls a, b si c  cand au terminat
2     operatia la banda. Daca au apasat butoanele doi dintre ei automatul
|     aprinde lampa de semnalizare L. Daca au apasat trei automatul porneste
|     motorul M care avanseaza banda cu o pozitie.
|     Contactul t al traductorului de pozitie se actioneaza cand banda
3     a ajuns in noua pozitie. Automatul opreste lampa L si motorul M si
|     asteapta noile impulsuri a, b si c de la muncitori.
|    t         a                                                    Ka
4---|/|-------| |-----------------------------------------------------(RLY)|
|         |    Ka   |
|         +---| |---+
|    t         b                                                    Kb
5---|/|-------| |-----------------------------------------------------(RLY)|
|         |    Kb   |
|         +---| |---+
|    t         c                                                    Kc
6---|/|-------| |-----------------------------------------------------(RLY)|
|         |    Kc   |
|         +---| |---+
|     Se memoreaza impulsurile a, b si c trimise automatului de catre
7     muncitorii la banda rulanta.
```


Ka
Kb
Kc
KFM
(RLY)
KFM
M
(OUT)
Filtrul logic al motorului care deplaseaza banda din hala de productie
Acesta trebuie sa porneasca atunci cand toti cei trei muncitori au
apasat pe butoanele a, b si c.
Ka
Kb
KFL
(RLY)
Kc
Kb
Kc
KFM
D
(TIM)
KFL
L
(OUT)
D
Filtrul logic al lampii de semnalizare care se aprinde atunci cand
doi muncitori au apasat pe butoanele a, b, c sau cand banda este in
repaus mai mult de 100 secunde (KFM neactionat si releul de timp D
actionat).

10.5.10. Simulatorul Tl5 pentru Windows

Modul de operare a acestui simulator pe calculatorul PC local este intuitiv şi asemănător cu varianta DOS Tl4. Varianta educativă a acestui simulator nu permite folosirea simulatorului sub forma unui server pentru controlarea la distanţă unui APL, pe intranet şi internet.

La fel ca şi la varianta Dos apare o dificultate la elaborarea programului sub formă de schemă desfăşurată (LD) atunci când se introduce o ramură paralelă care conţine mai multe contacte serie. În această variantă se folosesc contacte de tipul 5 la care trebuie să se stabilească distinct cele două puncte de conectare. Pentru aceasta se observă că atunci când se selectează un punct cu ajutorul mauselui sau a săgeţilor apare borna contactului sub forma unui pătrat sau punctul de conexiune sub forma unei pete. După selectarea unui contact cu mauseul transformarea bornei în punct de conexiune cu ajutorul tastelor săgeţi se face conectarea unui contact de tip 5 iar punctul de conexiune opus se transformă într-un x. Punctele de conexiune se găsesc pe ramura principală a circuitului pe care se găseşte bobina. După apariţia primei conexiuni sub forma unui x se determină pe circuitul principal cu ajutorul tastelor săgeţi al doilea punct de conexiune , care apare sub forma unei pete şi se conectează din nou componenta 5.

10.6. Simulatorul APL Easy Relay

Varianta demo a simulatorului poate fi descărcată de la adresa:
http://www.moeller.net/en/industry/switchgear/switch_control/easy/index.jsp

Programul are o documentaţie extinsă în limba română: Ajutor, Interfaţă utilizator, Curs introductiv.

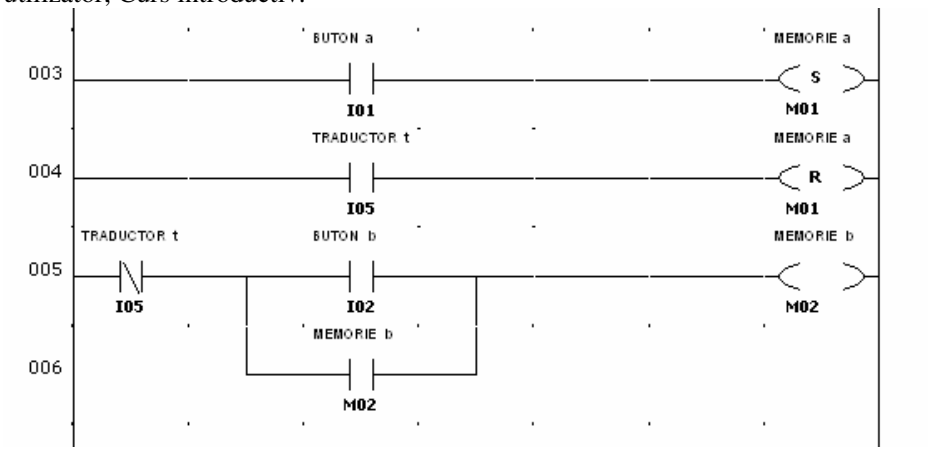


Fig. 10.13 Un exemplu de program pentru Easy Relay tip schema desfăşurată (LD)

Utilizarea sa este similară cu cea a simulatorului TL5 şi exemplul din paragraful 10.5.9 poate fi transpus uşor pentru Easy Relay. Un fragment de program este prezentat în Fig. 10.13.

11. Bibliografie

[1] Asociația inginerilor instalatori din România**, Manualul inginerului instalator**, Artecno, Bucureşti, 2002
[2] Astrom K. J**., Control System Design**, Lund Institute of Technology, 2001.
[3] Astrom K. J**., Model Uncertainty and Robust Control Design**, COSY Valencia Workshop, Sept., 1999.
[4] Bagh C**., Major systems theories throughout the world**, Behavioral Science, Apr.90, Vol. 35, Issue 2, p. 79-108
[5] Balan St., Larionescu S., **Modele fizice**, Institutul de Construcţii Bucureşti, 1974.
[6] Becerra V. M., **Robust Control**, DCMS, 2002
[7] Becerra V. M., **Advanced System Identification**, 2002.
[8] Caluianu S., **Inteligența artificială în instalații. Logica fuzzy şi teoria posibilităților**, Ed. MATRIX ROM, Bucureşti, 2000
[9] Cannon H., R., **Dynamics of physical systems**, McGraw-Hill, N.Z., 1967
[10] Chapurlat V., Prunet F., **Presentation du GRAFCET**: GRAphe Fonctionnel de Commande Etape – Transition, www.eerie.fr/~chapurla.
[11] Diatcu E., **Elemente fundamentale ale teoriei sistemelor şi calculatoarelor**. Ed. Hyperion XXI, Buc., 1997.
[12] Dorf R., C., Bishop R., H., **Modern Control Systems**, Addison-Wesley, New York, 1998.
[13] Drakos K., Ercolani M., **Introduction to Time Series Analysis**, University of Essex, Colchester UK, 2003.
[14] Dumitrache I., **Automatizări electronice**, Ed. didactică şi pedagogică, Buc., 1993.
[15] Dumitrache I., Constantin N., Drăgoicea M., **Rețele neurale**. Matrix Rom, Bucureşti, 1999.
[16] Dutton K., Thompson S., Barraclough., **The art of control engineering**, Addison-Wesley, New York, 1997.
[17] Eykhoff P., **Identificarea sistemelor**. Estimarea parametrilor şi stărilor pentru sistemele tehnice, economice, biologice, Ed., tehnică, Buc., 1977.
[18] Filip F., G., Bărbat B., **Informatica industrială**. Noi Paradigme şi aplicații, Ed. Tehnică, Buc., 1999.
[19] Goodwin G. C., Graebe S. F., Salgado M. E., **Control System Design,** Prentice Hall, N.Y., 2000.
[20] Ionescu C., Ionescu D., Larionescu S., Bădescu Gr., Ivan N., **Automatizarea instalațiilor.** Îndrumător de proiectare, Vol 1 şi 2, ICB, 1982.
[21] Ionescu C., Larionescu S., **Automatizări. Sisteme Automate Discrete Logice**, UTCB, 1997.
[22] Ionescu C., Vlădeanu V., Larionescu S., Ionescu D., **Automatizări**, Ed. didactică şi pedagogică, Bucureşti, 1982.
[23] Ionescu G., Ionescu V., **Automatica de la A la Z**, Ed. ştiințifică şi enciclopedică, Bucureşti, 1987.
[24] Ionescu C., Larionescu S., Caluianu S., Popescu D., **Automatizarea instalațiilor. Comenzi automate**, Ed. MATRIX ROM, Bucureşti, 2002.

[25] Jack H., **Automating Manufacturing Systems with PLC**, 2001.
[26] Kwakernaok H., Meinsma G., **Design Methods for Control Systems**, Dutch Institute of Systems and Control, 2001.
[27] Landau I., D., **Identificarea şi comanda sistemelor**, Ed. tehnică, Bucureşti, 1997.
[28] Larionescu S., **Accente noi în analiza şi proiectarea sistemelor automate**, A XXXVI-a Conferinţă naţională de instalaţii, Sinaia, 2-5 oct.2001, Vol. 2,p.57-65
[29] Larionescu S., **Aprecierea robusteţii sistemelor automate**, Măsurări şi Automatizări, Nr. 2, 2001, p.55-56.
[30] Larionescu S., **Aspecte moderne în proiectarea inginerească a sistemelor automate**, A XXXVII-a Conferinţă naţională de instalaţii, Sinaia, 1-4 oct., 2002, p.5-18.
[31] Larionescu S., **KitSAS – Simularea şi Analiza Sistemelor**, Matrix Rom, Buc., 1999
[32] Larionescu S., **KitSAS** v. 6.12, http://www.geocities.com/larionescu/
[33] Larionescu S., **Modelele sistemelor**: modele virtuale, funcţionale. structurale şi matematice, http://www.geocities.com/larionescu/
[34] Larionescu S., **Schema bloc a unui sistem automat modern**, A XXXV-a Conferinţa de instalaţii, Sinaia, 3-6 oct. 2000, Vol. 2, p. 78-83.
[35] Larionescu S., **Sisteme fizice cu parametrii concentraţi**, UTCB, Fac. de Instalaţii, Conferinţa a VIII-a, Vol. III, ConsPress, Bucureşti, 2002, p. 65 – 72.
[36] Leţia T., S., Aştilean A., M., **Sisteme cu evenimente discrete**: modelare, analiză, sinteză şi control, Ed. Albastră, 1998.
[37] Licker P.S., **Fundamentals of System Analysis**. Boyd / Fraser Publishing Company, Boston, 1987.
[38] Mann H., **Course on multipole modeling, simulation and analysis of multidisciplinary systems**, http://icoszm.cvut.cz/course/
[39] Mann H., **Dynast**, http://icoszm.cvut.cz
[40] Păstrăvanu O., **Sisteme cu evenimente discrete**. Tehnici calitative bazate pe formalismul reţelelor Petri, Ed. Matrix Rom, Buc., 1997.
[41] Popescu D., **Teoria sistemelor automate**, Matrix ROM, Buc., 2000.
[42] Riviera D. E., **Introduction to System Identification**, 1998
[43] Sângeorzan D., **Regulatoare adaptive**, Editura Militară, Buc., 1992.
[44] Scilab Group, **Introduction to Scilab**, INRIA, Unite de recherche de Rocquencourt, Projjet Meta2, France, http://www-rocq.inria.fr/scilab/.
[45] Tertişco M., Stoica P., **Identificarea şi estimarea parametrilor sistemelor**, Ed. Academiei R.S.R., Buc., 1980.
[46] Thamm M. T., **Dynamic Models for Controller Design**, University of Newcastle upon Tyne, 1999.
[47] Zadeh L., A., Polak E., **Teoria sistemelor**, Editura tehnică, Buc., 1969.